STRUCTURED SPARSITY: THEOREMS, ALGORITHMS AND APPLICATIONS

BY JUNZHOU HUANG

A dissertation submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Computer Science

Written under the direction of
Professor Dimitris N. Metaxas
and approved by

New Brunswick, New Jersey
October, 2011
Structured Sparsity: Theorems, Algorithms and Applications

by Junzhou Huang

Dissertation Director: Professor Dimitris N. Metaxas

Today, sparsity techniques have been widely used to address practical problems in the fields of medical imaging, machine learning, computer vision, data mining, compressive sensing, image processing, video analysis and multimedia. We will briefly introduce the related sparsity techniques and their successful applications on compressive sensing, sparse learning, computer vision and medical imaging. Then, we propose a new concept called strong group sparsity to develop a theory for group Lasso, which shows that group Lasso is superior to standard Lasso for strongly group-sparse data. It provides a convincing theoretical justification for using group sparsity regularization when the underlying group structure is consistent with the data. Moreover, the theory also predicts the limitations of the group Lasso formulations. To address those limitations, we further build a new framework called structured sparsity, which is a natural extension of the standard sparsity concept in statistical learning and compressive sensing. By allowing arbitrary structures on the feature set or data, this concept generalizes the group sparsity idea. A general theory (Group-RIP) is developed for learning with structured sparsity, based on the notion of coding complexity associated with the structure, which guarantees better performance with more structure information other than pure sparsity. The new sparsity techniques under this framework have been successfully applied to different applications, such as compressive sensing MR imaging, video background subtraction, object tracking in visual surveillance, dynamic scene registration, automatic image annotation, medical image analysis, fast tag separation in cardiac tMRIs, cervigram image segmentation and so on. The improved experimental results in these applications further validate the nice theoretical guarantees of
our new theorems and demonstrate the effectiveness of our new framework in the practical applications involved large scale data.
There were many people who helped me during my PhD studying career, and I would like to take this opportunity to thank them.

I am extremely grateful to my adviser, Prof. Dimitris Metaxas, for his guidance, support and encouragement throughout my Ph.D. years. He gave me the great freedom to pursue independent work, and also the incredible guidance to direct me toward the exciting application areas in our field. He taught me how to think about problems in a way that challenges me and helps me succeed. None of the work in this thesis would have happened without him.

I want to thank my co-advisor, Prof. Tong Zhang, who has been working closely with me and contributed numerous ideas and insights to our theoretical work on structured sparsity. He always made time for me to talk about ideas and discuss problems. He held me to the highest of standards, but also had the faith that I would be able to achieve them. His insightful comments and constructive criticisms have been invaluable to me.

My sincere thanks go to my thesis committee members, Prof. Vladimir Pavlovic, Prof. Tina Eliassi-Rad, Prof. Takeo Kanade, and my qualifying exam committee members, Prof. Ulrich Kremer, Prof. Michael Fredman, for their valuable suggestions regarding my early proposal and this thesis. It is a privilege for me to have each of them serve in my committees.

The research described in this thesis has benefitted from my other collaborators besides my advisors. Without them, some of the chapters in this thesis would not have been possible. My special thanks go to Prof. Leon Axel and Prof. Xiaolei Huang. I have been learning a lot from them through the collaborations.

I want to thank all my colleagues from the Center for Computational Biomedicine Imaging and Modeling (CBIM), the Computer Science Department, and the Biomedical Engineering Department. It is my pleasure to meet such a concentration of creative and nice people here. I am grateful to all with whom I spent my time as a graduate student at Rutgers.

Finally, my special thanks go to my family. I would like to express my earnest gratitude to my parents and my sisters for their love and countless sacrifices to give me the best possible education. I cannot thank my parents-in-law enough for giving me the opportunity to freely pursue my interests and career. Without their patience and unreserved support, it would not
have been possible to reach this stage in my career. I owe a debt of gratitude to my wife Xueqing who have loved and supported me throughout this thesis and throughout my life. This thesis is built on the sacrifice of her career in the bank. She made my life is never entirely consumed by research. Finally, thanks to my little angels Neo and Nicole, for the energy and the happiness they bring to me.
Dedication

To my wife Xueqing, my son Neo and my daughter Nicole, for their endless trust, support, encouragement and love
# Table of Contents

Abstract ......................................................... ii
Acknowledgements ........................................ iv
Dedication ..................................................... vi

List of Tables ................................................ x

List of Figures ................................................ xii

1. Introduction .............................................. 1
    1.1. Problem .................................................. 1
    1.2. Standard Sparsity ....................................... 3
    1.3. Related Work ........................................... 5
    1.4. Motivations ............................................. 8
    1.5. Organization ........................................... 9

2. The Benefit of Group Sparsity ............................. 12
    2.1. Introduction ............................................ 12
    2.2. Strong Group Sparsity ................................... 13
    2.3. Related Work ........................................... 14
    2.4. Assumptions ............................................ 15
    2.5. Main Results ........................................... 17
    2.6. Experiments ............................................ 20
      2.6.1. Even group size .................................... 20
        With correct group structure .......................... 20
        With incorrect group structure ....................... 22
      2.6.2. Uneven group size .................................. 22
    2.7. Discussions ........................................... 24
    2.8. Proofs ................................................ 25
      2.8.1. Proof of Proposition 2.4.1 ........................ 25
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.8.2. Proof of Proposition 2.4.2</td>
<td>26</td>
</tr>
<tr>
<td>2.8.3. Technical Lemmas</td>
<td>29</td>
</tr>
<tr>
<td>2.8.4. Proof of Theorem 2.5.1</td>
<td>33</td>
</tr>
<tr>
<td>3. Learning With Structured Sparsity</td>
<td>35</td>
</tr>
<tr>
<td>3.1. Introduction</td>
<td>35</td>
</tr>
<tr>
<td>3.2. Related Work</td>
<td>36</td>
</tr>
<tr>
<td>3.3. Structured Sparsity</td>
<td>38</td>
</tr>
<tr>
<td>3.4. General Coding Scheme</td>
<td>39</td>
</tr>
<tr>
<td>3.5. Algorithms for Structured Sparsity</td>
<td>44</td>
</tr>
<tr>
<td>3.6. Theory of Structured Sparsity</td>
<td>45</td>
</tr>
<tr>
<td>3.6.1. Assumptions</td>
<td>45</td>
</tr>
<tr>
<td>3.6.2. Coding complexity regularization</td>
<td>46</td>
</tr>
<tr>
<td>3.6.3. Structured greedy algorithm</td>
<td>47</td>
</tr>
<tr>
<td>3.7. Experiments</td>
<td>49</td>
</tr>
<tr>
<td>3.7.1. Simulated 1D Signals with Line-Structured Sparsity</td>
<td>49</td>
</tr>
<tr>
<td>3.7.2. 2D Image Compressive Sensing with Tree-structured Sparsity</td>
<td>56</td>
</tr>
<tr>
<td>3.7.3. Background Subtracted Images for Robust Surveillance</td>
<td>58</td>
</tr>
<tr>
<td>3.8. Discussion</td>
<td>62</td>
</tr>
<tr>
<td>3.9. Appendix: Proofs of Theorems</td>
<td>64</td>
</tr>
<tr>
<td>3.9.1. Proof of Proposition</td>
<td>64</td>
</tr>
<tr>
<td>3.9.2. Proof of Theorem 2.5.1</td>
<td>65</td>
</tr>
<tr>
<td>3.9.3. Proof of Theorem 2.5.2</td>
<td>66</td>
</tr>
<tr>
<td>3.9.4. Proof of Theorem 2.5.3</td>
<td>69</td>
</tr>
<tr>
<td>3.9.5. Proof of Theorem 2.5.4</td>
<td>71</td>
</tr>
<tr>
<td>4. Applications in Compressive Sensing and Computer Vision</td>
<td>73</td>
</tr>
<tr>
<td>4.1. Introduction</td>
<td>73</td>
</tr>
<tr>
<td>4.2. Theory Review</td>
<td>75</td>
</tr>
<tr>
<td>4.3. Dynamic Group Sparsity</td>
<td>76</td>
</tr>
<tr>
<td>4.3.1. Dynamic Group Sparse Data</td>
<td>76</td>
</tr>
<tr>
<td>4.3.2. Dynamic Group Sparsity Recovery</td>
<td>78</td>
</tr>
<tr>
<td>4.4. DGS for Compressive Sensing</td>
<td>82</td>
</tr>
</tbody>
</table>
6.2.1. Problem ....................................................... 130
6.2.2. Method ......................................................... 132
6.2.3. Experiments .................................................. 134
6.2.4. Discussions .................................................. 135

6.3. Transformation-invariant Sparse Representation ..................... 136
   6.3.1. Problem ..................................................... 136
   6.3.2. Related Work ................................................. 138
       Convex optimization ............................................. 138
       Sparse Representation .......................................... 139
   6.3.3. Transform-invariant Sparse Representation ..................... 141
       Formulation ..................................................... 141
       Algorithm ..................................................... 141
       Simultaneous Face Alignment and Recognition ..................... 145
       Online Dynamic Texture Registration ......................... 146
   6.3.4. Experiments ................................................ 147
       Face Recognition ............................................... 148
       Dynamic Scene Registration ................................... 149
   6.3.5. Discussions ................................................ 151

6.4. Summary .......................................................... 152

7. Conclusions .......................................................... 153

References ............................................................ 157

Vita ................................................................. 166
List of Tables

5.1. Comparison of the SNR (db) over 100 runs ........................................ 115
5.2. Comparison of the CPU Time (s) over 100 runs .................................. 115
6.1. The average/maximum tracking error (pixels) .................................... 129
6.2. Performance comparison between 4 classifiers, measured by the mean of sensitivity and specificity. .................................................. 135
6.3. Identification rates under affine transformations .................................. 149
6.4. FEF of horizontal cumulative motion .................................................. 150
List of Figures

1.1. Two images from the movie “The Matrix”. With a glance, most of us will be
attracted by the young lady who wearing red clothes. .......................... 2
1.2. The evolution of prior models in the past 30 years. .............................. 3
1.3. Structured sparsity. (a) The brain image has tree sparsity after wavelet trans-
formation; (b) The background subtracted image has graph sparsity. ............. 9
2.1. Recovery results when the assumed group structure is correct. (a) Original data;
(b) results with Lasso (recovery error is 0.3444); (c) results with Group Lasso
(recovery error is 0.0419). .................................................... 21
2.2. Recovery performance: (a) recovery error vs. sample size ratio $n/k$; (b) recovery
error vs. group number $g$. .................................................. 21
2.3. Recovery results when the assumed group structure is incorrect. (a) Original
data; (b) results with Lasso (recovery error is 0.3616); (c) results with Group
Lasso (recovery error is 0.6688). .............................................. 23
2.4. Recovery performance: (a) recovery error vs. sample size ratio $n/k$; (b) recovery
error vs. group size $k_0$. ...................................................... 23
2.5. Recovery performance: (a) $g$ active groups have randomly uneven group sizes;
(b) half of $g$ active groups are single element groups and another half of $g$ active
groups have large group size. ............................................. 24
2.6. Recovery performance: (a) all $g$ active groups have large group size; (b) all $g$
active groups are single element groups. .................................... 25
3.1. Examples of structured sparsity, where black nodes are selected variables in all
available nodes: (a) standard sparsity; (b) group sparsity; (c) line-graph sparsity;
(d) tree sparsity and wavelet tree sparsity; (e) grid sparsity and graph sparsity
in background subtracted images. ........................................... 41
3.2. Recovery results of 1D signal with strongly line-structured sparsity. (a) original data; (b) recovered results with OMP (error is 0.9921); (c) recovered results with Lasso (error is 0.8660); (d) recovered results with Group Lasso (error is 0.4832 with group size gs=2); (e) recovered results with Group Lasso (error is 0.4832 with group size gs=4); (f) recovered results with Group Lasso (error is 0.2646 with group size gs=8); (g) recovered results with Group Lasso (error is 0.3980 with group size gs=16); (h) recovered results with StructOMP (error is 0.0246).

3.3. Recovery performance: (a) Recovery Error vs. Sample Size Ratio \((n/k)\); (b) CPU Time vs. Sample Size Ratio \((n/k)\).

3.4. Recovery performance in terms of Noise Levels: (a) Recovery Error vs. Noise Level; (b) CPU Time vs. Noise Level.

3.5. Performance Comparisons between methods related with structured sparsity (ModelCS [7], OverlapLasso [70], StructOMP). (a) Recovery Error vs. Sample Size Ratio \((n/k)\); (b) CPU Time vs. Sample Size Ratio \((n/k)\).

3.6. Recovery results of 1D weakly sparse signal with line-structured sparsity. (a) original data; (b) recovered results with OMP (error is 0.5599); (c) recovered results with Lasso (error is 0.6686); (d) recovered results with Group Lasso (error is 0.4732 with group size gs=2); (e) recovered results with Group Lasso (error is 0.2893 with group size gs=4); (f) recovered results with Group Lasso (error is 0.2646 with group size gs=8); (g) recovered results with Group Lasso (error is 0.5459 with group size gs=16); (h) recovered results with StructOMP (error is 0.0846).

3.7. Recovery performance for 1D Weak Line-Sparsity: (a) Recovery Error vs. Sample Size Ratio \((n/k)\); (b) CPU Time vs. Sample Size Ratio \((n/k)\).

3.8. Recovery results with sample size \(n = 2048\): (a) cameraman image; (b) recovered image with OMP (error is 0.1886; CPU time is 46.16s); (c) recovered image with Lasso (error is 0.1670; CPU time is 60.26s) and (d) recovered image with StructOMP (error is 0.0375; CPU time is 48.99s).

3.9. Recovery performance for 2D wavelet tree sparsity: (a) Recovery Error vs. Sample Size; (b) CPU Time vs. Sample size.

3.10. Recovery results with sample size \(n = 900\): (a) the background subtracted image; (b) recovered image with OMP (error is 1.1833); (c) recovered image with Lasso (error is 0.7075) and (d) recovered image with StructOMP (error is 0.1203).
3.11. Recovery performance. (a) Recovery Error vs. Sample Size; (b) CPU Time vs. Sample size. ........................................... 61

3.12. Recovery results with sample size $n = 600$: (a) the background subtracted image; (b) recovered image with Group OMP (error is 1.1340); (c) recovered image with Group Lasso (error is 0.6972) and (d) recovered image with StructOMP (error is 0.0808). ........................................... 62

3.13. Recovery performance: (a) Recovery Error vs. Sample Size; (b) CPU Time vs. Sample size. ........................................... 63

4.1. An example of dynamic group sparse data: (a) one video frame; (b) the foreground image; (c) the foreground mask and (d) the background subtracted image with the proposed algorithm. ........................................... 76

4.2. Main steps of the proposed DGS Recovery algorithm ........................................... 79

4.3. Recovery results of 1D data. (a) Original data; (b) GPSR (error is 0.5173 and time is 0.1847 seconds); (c) SPGL1-Lasso (error is 0.4021 and time is 1.1497 seconds); (d) OMP (error is 1.0270 and time is 0.1422 seconds); (e) SP (error is 0.6143 and time is 0.1100 seconds); (f) DGS recovery (error is 0.0178 and time is 0.0605 seconds). ........................................... 83

4.4. Recovery errors vs. measurement numbers: a) recovery errors; (b) running times. ........................................... 83

4.5. Recovery results of a 2D color image: (a) original color image; (b) recovered image with MCS [72] (error is 0.8399 and time is 29.2656 seconds); (c) recovered image with SP [29] (error is 0.7605 and time is 1.6579 seconds) and (d) recovered image with DGS (error is 0.1176 and time is 1.0659 seconds). ........................................... 84

4.6. Recovery performance: a) errors; (b) running times. ........................................... 85

4.7. Background subtraction in video surveillance where foreground is detected as anomaly. The columns of $A$ represent the previous background only images. ........................................... 87

4.8. Results on the Zhong’s dataset (a) original frame; (b) background subtracted image with AdaDGS; (c) the binary mask with AdaDGS and d) with robust Kalman model [156]. ........................................... 88

4.9. Results on the Elgammal’s dataset. (a) original frame; (b) with AdaDGS; (c) with KDE model [37] (d) with MoG [120]. ........................................... 89

4.10. Results on Monnet’s dataset. (a) original frame; (b) with AdaDGS; (c) with Monnet’s method [94] and (d) with MoG [120]. ........................................... 89

xv
4.11. Results on a helicopter video. (a) original frame; (b) with AdaDGS; (c) with MoG [120] and (d) with robust Kalman model [156]. ........................................ 89
4.12. The first stage: sparse representation and optimization for template tracking and occlusion estimation ................................................................. 93
4.13. The second stage: feature selection to accelerate the tracking process ........ 94
4.14. The tracking results of a car sequence in an open road environment. The vehicle was driven beneath a bridge which led to large illumination changes. Results from our algorithm, MIL, L1, and IVT are shown from top to bottom rows respectively. 96
4.15. The tracking results of a moving face sequence, which has large pose variation, scaling, and illumination changes. The order of the row sequences is the same as Figure 4.14. ................................................................. 97
4.16. The tracking results of a plush toy moving around under different pose and illumination conditions. The order of the rows is the same as in Figure 4.14. .... 98
4.17. The tracking results of the car sequence in a dark environment. This sequence has low resolution and poor contrast, which introduce some landmark ambiguity. Results of our algorithm, MIL, L1, and IVT are shown from top to bottom rows respectively. ................................................................. 99
4.18. The tracking results of a face sequence, which includes a lot of pose variations, partial or full occlusions. Results of our algorithm, MIL, L1, and IVT are shown from top to bottom rows respectively. ................................................................. 99
5.1. The MRI machines. ............................................................................ 100
5.2. The flow chart of compressive sensing MRI ......................................... 102
5.3. FCSA-MRI: the steps in each iteration. ............................................... 108
5.4. MR images: a) Cardiac; (b) Brain; (C) Chest and (d) Artery. ................. 109
5.5. Cardiac MR image reconstruction from 20% sampling (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 9.86, 14.43, 15.20, 16.46 and 17.57 (db). Their CPU time are 2.87, 3.14, 3.07, 2.22 and 2.29 (s). ....................... 110
5.6. Brain MR image reconstruction from 20% sampling (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 8.71, 12.12, 12.40, 18.68 and 20.35 (db). Their CPU time are 2.75, 3.03, 3.00, 2.22 and 2.20 (s). ....................... 111
5.7. Chest MR image reconstruction from 20% sampling (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 11.80, 15.06, 15.37, 16.53 and 16.07 (db). Their CPU time are 2.95, 3.03, 3.00, 2.29 and 2.234 (s).

5.8. Artery MR image reconstruction from 20% sampling (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 11.73, 15.49, 16.05, 22.27 and 23.70 (db). Their CPU time are 2.78, 3.06, 3.20, 2.22 and 2.20 (s).

5.9. Performance comparisons (CPU-Time vs. SNR) on different MR images: a) Cardiac image; (b) Brain image; (c) Chest image and (d) Artery image.

5.10. Full Body MR image reconstruction from 25% sampling (a) Original image; (b), (c), (d) and (e) are the reconstructed images by the TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 12.56, 13.06, 18.21 and 19.45 (db). Their CPU time are 12.57, 11.14, 10.20 and 10.64 (s).

5.11. Performance comparison on the full body MR image with different sampling ratios. The sample ratios are: (1) 36%; (2) 25% and (3) 20%. The performance: (a) Iterations vs. SNR (db) and (b) Iterations vs. CPU Time (s).

6.1. Tagged cardiac MR images and their corresponding magnitude spectrums

6.2. A cardiac tagged MR image can be decomposed into a tag-only image and a piecewise-smooth cardiac image without tag.

6.3. Tag separation. (a) the input image \(I\); (b) the separated tag \(T\); (c) the piecewise smooth image \(S\); (d) noise.

6.4. Comparisons. (1) visual comparisons in the whole images; (2) and (3) visual comparisons in the local regions; (4) segmentation comparisons; (a) tagged images; (b) the band-pass method [91]; (c) the band-stop method [105]; (d) the proposed approach.

6.5. Tag tracking comparisons. (a) and (c) mis-track occurs in the areas indicated by the green circles in the original tagged image sequence; (b) and (d) the corresponding results by our approach.
Examples of digitized cervicographic images (i.e. cervigrams) of the uterine cervix, created by the National Library of Medicine (NLM) and the National Cancer Institute (NCI). Ground truth boundary markings by 20 medical experts. Our work is aimed for automatically segmenting the biomarker AcetoWhite (AW) regions, which indicates clinical significance.

Color distribution of AW (red) and non-AW (blue) regions. The left one comes from one sample. The right one comes from one hundred samples.

Results of different algorithms applied on two images. 1st column: original image; 2nd column: ground truth; 3rd column: SVM with radial basis function kernel; 4th column: nearest neighbor; 5th column: reconstructive sparse representations.

Sparse representation results on aligned images. (a) One of training images; (b) the test image; (c) the linear sparse representation result [141] and the sparse representation result using the proposed method.

Sparse representation results given unaligned test images. Training images (a), test images (b), results by Randomfaces [141] (c), results by the proposed approach.

Identification results and ROC curves.

The Escalator Sequence [156].

The cumulative motion estimation.

A sequence of a moving flower bed [40, 106, 135, 58].
Chapter 1

Introduction

The data arising from many real-world domains, such as emails, images, videos, and social media, are usually high-dimensional, richly structured and heterogeneous. These data only become useful for us if we are able to quickly access those pieces that we are really interested in. Generally, our interested data pieces are far less than the whole data. Thus sparsity, the notion that only a few items are important out of many, often helps us for data analysis as an useful prior model. In our attempts to find useful information from numerous data around us we are always trying to use the sparsity to help us.

As scientists and researchers, we are often faced with large and complicated data. These data only become tractable if we are able to extract their structure properties that characterize the entire data. Thus structure description can determinate whether the data can be efficiently modeled or is hopelessly complex. In our attempts to understand the complex data in the practical applications we are always trying to simplify and reduce to essentials. Therefore, we are always on the way to look for structures.

One key point of this thesis is thus to integrate the structure and sparsity into one framework to explore data efficiently and effectively in practice. Structured Sparsity is thus the key theme of this thesis. More specifically, we will build a general framework of structured sparsity which has theoretical benefits and practical advantages over previous tools.

1.1 Problem

Our story begins with a question. How effectively we can extract useful information from massive data in limited time? One paper published in nature journal gives a possible solution to this question in 1996 [102]: human brain uses sparsity as an effective tool to extract information from outside sources, because the interested information is always sparse compared with the whole data. For example, Figure 1.1 shows two images from the movie “The Matrix”. What is ones first impression after seeing these images? Yes, the answer is that the young lady wearing red clothes. Although there are a lot of walking people who wearing dark clothes, most of
us are impressed by this young lady with red clothes. Our brains use sparsity techniques to automatically ignore other people and quickly extract the discriminant information in these images.

Figure 1.1: Two images from the movie “The Matrix”. With a glance, most of us will be attracted by the young lady who wearing red clothes.

What is sparsity? Here is the mathematical definition of the sparsity. We say data $x$ with dimension $p$ is sparse if only $k$ out of $p$ coordinates are nonzeros and the sparsity number $k$ is far less than its dimension $p$. In practice, most of data are not exactly sparse but compressible. It means that these data can be well approximated by the $k$-sparse data. Actually, the key idea of sparsity has been widely used for image compression, such as JPEG and JPEG 2000. As we know, the wavelet coefficients of an image are compressible and can be well approximated by the sparse data. The sparsity prior model can be used to efficiently recover the original image if necessary.

It is a dominant theme to design the prior model in modern science, such as physics, statistics, machine learning and so on. It is generally guided by some prior knowledge on what the solution should be like. The underlying idea of the prior models is to bias estimation problems towards preferred solutions. Here, the notion of preference depends on some prior knowledge of what the solution should be in practice. It is able to avoid overfitting when little data is available compared with the complexity of the problem. This also enables to transfer an ill-posed problem to a well-posed problem. A well-posed problem [123] is generally defined as: 1) a solution exists; 2) the solution is unique; and 3) the solution depends continuously on the data, in some reasonable topology. One well known prior model is introduced by A. N. Tikhonov in [123]. The idea was to add a penalty term in the formulation to penalize the Hilbertian norm of the loss function, which can improve the conditioning of the problem. Figure 1.2 shows the evolution of the prior models in the past 30 years. The first one is the energy model, which has been widely used in different fields [123]. The second is robust statistics model [69]. It is more robust for the practical data with outliers. Total Variation and Wavelet are two well-known
tools [88, 81]. They are very popular in the past 20 years. The Total Variation model prefers to recover solutions with sparse gradients. The Wavelet model prefers to obtain the piecewise-smooth solutions. Standard sparsity became very popular since 2006 [20, 21, 32]. Unlike the Wavelet model with a fixed basis, it is data driven and the sparsity basis $D$ can be adaptively learned from the data for different applications, which is more flexible. In practical data, we always know more structured priors other than pure sparsity. My thesis work focus on combining the sparsity prior with other structure priors together under a unified framework. We call it structured sparsity. The advantages of structured sparsity will be theoretically proved in the following chapters.

$$g(x) = \lambda \| Lx \|_2$$

$$g(x) = \lambda \rho \left\{ Lx \right\}$$

$$g(x) = \lambda \| \nabla x \|_1$$

Figure 1.2: The evolution of prior models in the past 30 years.

1.2 Standard Sparsity

The standard sparsity has been widely used for different fields because of its effectiveness and flexibility. It is the key tool for compressive sensing and sparse learning. We detail its applications on compressive sensing and sparse learning respectively in the following.

Compressive Sensing

The compressive sensing (CS) theory has shown that a sparse signal can be recovered from a small number of its linear measurements with high probability [21, 32]. According to CS, a
sparse signal \( x \in \mathbb{R}^p \) should be recovered from the following random linear projections:

\[
y = \Phi x + e,
\]

where \( y \in \mathbb{R}^n \) is the measurement vector, \( \Phi \in \mathbb{R}^{n \times p} \) is the random projection matrix, \( n \ll p \), and \( e \) is the measurement noise. The CS theory is magnetic as it implies that the signal \( x \in \mathbb{R}^p \) can be recovered from only \( n = \mathcal{O}(k \log(p/k)) \) measurements [21] if \( x \) is a \( k \)-sparse signal, which means that \( x \in \mathbb{R}^p \) can be well approximated using \( k \ll p \) nonzero coefficients under some linear transformations. It directly leads to the potential of cost saving in digital data capturing. Although the encoding in data capturing only involves simple linear projections, signal recovery requires nonlinear algorithms to seek the sparsest signal from the measurements. This problem can be formulated with \( l^0 \) minimization:

\[
x_0 = \arg\min ||x||_0 \quad \text{while} \quad ||y - \Phi x||_2^2 < \varepsilon
\]

where \( || \cdot ||_0 \) denotes the \( L_0 \)-norm that counts the number of nonzero entries and \( \varepsilon \) is the noise level.

**Sparse Learning**

Consider a fixed set of \( p \) basis vectors \( \{x_1, \ldots, x_p\} \) where \( x_j \in \mathbb{R}^n \) for each \( j \). Denote by \( X \) the \( n \times p \) data matrix, with column \( j \) of \( X \) being \( x_j \). Given a random observation \( \bar{y} = [y_1, \ldots, y_n] \in \mathbb{R}^n \) that depends on an underlying coefficient vector \( \bar{\beta} \in \mathbb{R}^p \), we are interested in the problem of estimating \( \bar{\beta} \) under the assumption that the target coefficient \( \bar{\beta} \) is sparse. For sparse learning, we consider fixed design only. That is, we assume \( X \) is fixed, and randomization is with respect to the noise in the observation \( y \).

We consider the situation that the expectation \( \mathbb{E}y \) can be approximated by a sparse linear combination of the basis vectors:

\[
\mathbb{E}y \approx X \bar{\beta},
\]

where we assume that \( \bar{\beta} \) is sparse. Define the support of a vector \( \beta \in \mathbb{R}^p \) as all the indices of non-zero elements

\[
\text{supp}(\beta) = \{j : \beta_j \neq 0\},
\]

and \( ||\beta||_0 = |\text{supp}(\beta)| \). A natural method for sparse learning is \( L_0 \) regularization:

\[
\hat{\beta}_{L_0} = \arg\min_{\beta \in \mathbb{R}^p} \hat{Q}(\beta) \quad \text{subject to} \quad ||\beta||_0 \leq s,
\]

where \( s \) is the desired sparsity. For simplicity, unless otherwise stated, the objective function considered throughout this chapter is the least squares loss

\[
\hat{Q}(\beta) = ||X\beta - y||_2^2,
\]
Other objective functions for generalized linear models (such as logistic regression) can be similarly analyzed.

From above introductions, we can know that the compressive sensing and sparse learning have similar formulations. Both of them are based on standard sparsity. The problems are NP-hard. In general, no known procedure can correctly find the sparsest solution more efficiently than exhausting all subsets of the entries for $x$ or $\beta$. The key problem in CS and sparse learning is thus to develop efficient optimization algorithms with nearly optimal theoretical performance guarantees. Since they are generally NP-hard, in practice, one often considers approximate solutions. A standard approach is convex relaxation of $L_0$ regularization to $L_1$ regularization, often referred to as Lasso [122]. Another commonly used approach is greedy algorithms, such as the orthogonal matching pursuit (OMP) [127].

In practical applications, one often knows a structure on the coefficient vector $x$ or $\beta$ in addition to sparsity. For example, in group sparsity, one assumes that variables in the same group tend to be zero or nonzero simultaneously. The purpose of this thesis is to study the general estimation problem under structured sparsity. If meaningful structures exist, we show that one can take advantage of such structures to improve the standard sparsity model.

### 1.3 Related Work

There are a lot of algorithms, which have been developed for sparsity. Generally, they can be classified into three classes.

Convex relaxation is a well-known class of algorithms for sparsity. A standard approach is convex relaxation of $L_0$ regularization to $L_1$ regularization, often referred to as Lasso [122]. To accelerate convergence, a technique based on Barzilai-Borwein (BB) steps is used. Hale et al. propose a fixed-point continuation (FPC) method [52], which is based on operator splitting, to solve the more general problem. They established a Q-linear rate of convergence of the method without assuming strict convexity nor uniqueness of solution. Wright et al. [108] propose an iterative (called SpaRSA) method, in which separable (but not necessarily smooth nor convex) regularization term is used instead of $L_1$ norm. Pareto root-finding approach is used to perform the $L_1$ regularization optimization in SPGL1 [13], which is efficiently solved with the spectral gradient projection. In [151], a block coordinate gradient descent method is proposed to solve the $L_1$ regularized problems. The Q-linear convergence rate can be obtained when the coordinate block is chosen by a Gauss-Southwell-type rule. Zhang [153] proposes a two-stage relaxation scheme for the capped-L1 regularization with the theoretical analysis.
Iterative greedy pursuit is another well-known class of algorithms for sparsity. The earliest ones include the matching pursuit [90] and orthogonal matching pursuit (OMP) [127]. Their successors include the stagewise OMP [34] and the regularized OMP [98]. The subspace pursuit [29] and the compressive sampling matching pursuit (CoSaMP) [97] are proposed recently by incorporating backward steps. They have similar theoretical recovery guarantees as that of the Basis Pursuit methods, while their computation complexity is comparable to those of the greedy pursuit algorithms. Zhang [152] recently proposes a combination algorithm that is based on the forward greedy algorithm but takes backward steps adaptively whenever beneficial.

The sparsity problem can also be formulated in a Bayesian framework. In [73], the relevance vector machine (RVM) [124] is adapted to this problem. Independent priors are utilized for each coefficient in an expectation-propagation framework in [112]. They consider both signal reconstruction and measurement design problems. However, all required parameters are not estimated, but rather left as parameters to be tuned. In [111], the sparsity is modelled with the Laplace priors on the basis coefficients in a hierarchical manner. Its formulation includes the RVM formulation as a special case. All required model parameters are estimated along with the unknown sparse coefficients. Belief Propagation Decoding is used to approximate the Bayesian inference in [10], where the data matrix is represented as a graphical model. In [109], the minimum mean squared error (MMSE) estimation is used as the optimization metric to obtain robust solution under the Bayesian framework.

All of these algorithms do not consider sparse data structures other than sparsity. However, in some practical applications, the sparse coefficients are often dependent and structured. In these cases, it is thus emerged how to combine the data structure and the sparsity together to obtain better results.

The idea of using structure in addition to sparsity has been explored before. An example is the group structure, which has received much attention recently. For example, group sparsity has been considered for simultaneous sparse approximation [140] and multi-task compressive sensing [72] from the Bayesian hierarchical modeling point of view. Under the Bayesian hierarchical model framework, data from all sources contribute to the estimation of hyper-parameters in the sparse prior model. The shared prior can then be inferred from multiple sources. He et al. recently extend the idea to the tree sparsity in the Bayesian framework [55, 54]. Although the idea can be justified using the standard Bayesian intuition, there are no theoretical results showing how much better (and under what kind of conditions) the resulting algorithms perform. In the statistical literature, Lasso has been extended to the group Lasso when there exist group/block structured dependency among the sparse coefficients [148]. Recently, a more
efficient spectral projected-gradient algorithm [14] is proposed to solve the group Lasso problem, which is based on a linear-time algorithm for Euclidean projection. The algorithm is more suitable for large scale group sparse learning. Elitist Lasso is proposed to the same problem with the $L_{1,2}$ norm regularization in [78].

However, none of the above mentioned work was able to show advantage of using group structure. Although some theoretical results were developed in [6, 96], neither showed that group Lasso is superior to the standard Lasso. The authors of [76] showed that group Lasso can be superior to standard Lasso when each group is an infinite dimensional kernel, by relying on the fact that meaningful analysis can be obtained for kernel methods in infinite dimension. In [101], the authors consider a special case of group Lasso in the multi-task learning scenario, and show that the number of samples required for recovering the exact support set is smaller for group Lasso under appropriate conditions. In [65], a theory for group Lasso was developed using a concept called strong group sparsity, which is a special case of the general structured sparsity idea considered here. It was shown in [65] that group Lasso is superior to standard Lasso for strongly group-sparse signals, which provides a convincing theoretical justification for using group structured sparsity.

While group Lasso works under the strong group sparsity assumption, it could not handle general structures in practice. Several limitations of group Lasso were mentioned in [65]. For example, group Lasso does not correctly handle overlapping groups (in that overlapping components are over-counted); that is, a given coefficient should not belong to different groups. This requirement is too strict for many practical applications. To address this issue, a method called composite absolute penalty (CAP) is proposed in [155] which can handle overlapping groups. Unfortunately, no theory is established to demonstrate the effectiveness of the approach. In a related development [80], Kowalski et al. generalized the mixed norm penalty to structured shrinkage, which can identify the structured significance maps and thus can handle the case of the overlapping groups, However, the structured shrinkage operations do not necessarily convergence to a fixed point. There were no additional theory to justify their methods.

Different regularization terms are proposed to extend the group Lasso, such as elastic net penalty [158], composite absolute penalty (CAP) [155] and mixed norm penalty [148, 78, 77, 79, 84]. The hierarchy structure can be well exploited by a mixed norm penalty [77, 79, 84]. One of their shortcomings is the independence of the static groups. It means that a given coefficient can not belong to different groups, which make them too rigid for some practical data. In [155], composite absolute penalty is proposed to handle both cases of none-overlapping groups and overlapping groups. However, no theoretical proof is given to the CAP.
Tropp et al. proposed the simultaneous OMP for group sparsity problem in [128]. In [126], the convex relaxation solution is proposed for the same problem. The theoretical analysis is also provided. However, they assume that the group structures are known before recovery. Moreover, they only consider the case where all groups share a common nonzero coefficient support set. In [121], the authors attempt to derive a bound on the number of samples needed to recover block sparse signals, where the sparse coefficients in each block are either all zero or all nonzero. This corresponds to the case of group sparsity with equal size groups. It is not clear how to extend their work to the general structured sparsity.

Other structures have also been explored in the literature. For example, tonal and transient structures were considered for sparse decomposition of audio signals in [30], but again without any theory. Grimm et al. [50] investigated positive polynomials with structured sparsity from an optimization perspective. The theoretical result there did not address the effectiveness of such methods in comparison to standard sparsity. Baraniuk et al. [7] introduced a specific case of structured sparsity, referred to as model based sparsity. It is important to note that some theoretical results were obtained there to show the effectiveness of their method in compressive sensing. In [19], the idea has been successfully applied to a data model motivated by neuronal spike trains. However, they do not provide a generic framework for structured sparsity. Thus, different schemes have to be specifically designed for different data models. It remains as an open issue how to develop a general theory for structured sparsity, together with a general structure representation and algorithm that can be applied to a wide class of such problems.

1.4 Motivations

From introduction above, we know that the standard sparsity model only captures simple primary data structure. However, practical data generally has more structure information than just sparsity. With more data structure priors, we expect to obtain better results with less measurement requirement and computation complexity. Here are two examples for structured sparsity shown in Figure 1.3. Figure 1.3(a) shows an example of tree sparsity. The brain image has tree sparsity after wavelet transformation. In this case, if one node has a non-zero value in a tree, then its parent-node or all ancestor nodes should have non-zero values too. Figure 1.3(b) shows an example of graph sparsity. The background subtracted image has graph sparsity. In a graph, if one node has a non-zero value, then its neighboring nodes should have non-zero values too with high probabilities. This is based on the fact that the interested foreground objects often occur a group of clustered pixels.
From above introductions, we know that there exists literature on structured sparsity, with empirical evidence showing that one can achieve better performance by imposing additional structures. However, none of the previous work was able to establish a general theoretical framework for structured sparsity that can quantify its effectiveness. The goal of this thesis is to develop such a general theory that addresses the following issues, where we pay special attention to the benefit of structured sparsity over the standard non-structured sparsity:

- Quantifying structured sparsity;
- The minimal number of measurements required in compressive sensing;
- Estimation accuracy under stochastic noise;
- An efficient algorithm that can solve a wide class of structured sparsity problems.

1.5 Organization

We now outline the theoretical framework and application results which are considered by subsequent thesis chapters. The brief introductory paragraphs provide more detailed outlines for each chapters.

Chapter 2 investigates the benefits of the sparsity with group structure. As we know, Group Lasso is a well-known algorithm for group sparsity in statistical learning. A theory is developed in this chapter for Group Lasso using a concept called strong group sparsity. We theoretically prove that group Lasso is superior to standard Lasso for strongly group-sparse data. This provides a convincing theoretical justification for using group sparsity regularization when the underlying group structure is consistent with the data. Moreover, our theory can also help predict some limitations of the group Lasso formulation. We conduct a series of simulated experiments to validate the benefits and limitations of the group sparsity.
Chapter 3 investigates a new learning formulation called structured sparsity, which is a natural extension of the standard sparsity concept in statistical learning and compressive sensing. By allowing arbitrary structures on the feature set, this concept generalizes the group sparsity idea. A general theory Group-RIP is developed for learning with structured sparsity, based on the notion of coding complexity associated with the structure. It is shown that if the coding complexity of the target signal is small, then one can achieve improved performance by using coding complexity regularization methods, which generalize the standard sparse regularization. Moreover, a structured greedy algorithm is proposed to efficiently solve the structured sparsity problem. Experiments are conducted to demonstrate the advantage of structured sparsity over standard sparsity on some real applications.

Chapter 4 gives the applications of structured sparsity in compressive sensing and computer vision. The nonzero coefficients are often not random but tend to be clustered in some practical sparse data. Intuitively, better results can be achieved in these cases by reasonably utilizing both clustering and sparsity priors. Motivated by this idea, we developed a new greedy sparse recovery algorithm, which prunes data residues in the iterative process according to both sparsity and group clustering priors rather than only sparsity as in previous methods. In compressive sensing, the proposed algorithm can recover stably sparse data with clustering trends using far fewer measurements and computations than current state-of-the-art algorithms with provable guarantees. Moreover, our algorithm can adaptively learn the dynamic group structure and the sparsity number if they are not available in the practical applications, such as video background tracking and visual tracking and etc.

Chapter 5 proposes an efficient algorithm for fast image reconstruction for compressed MR imaging based on structured sparsity. The problem is formulated as the minimization of a linear combination of three terms corresponding to a least square data fitting, total variation (TV) and $L_1$ norm regularization. An new algorithm, Fast Composite Splitting Algorithm (FCSA), is developed to efficiently solve this problem. First, the original problem is decomposed into $L_1$ and TV norm regularization subproblems respectively. Then, these two subproblems are efficiently solved by existing techniques. Finally, the reconstructed image is obtained from the weighted average of solutions from two subproblems in an iterative framework. The proposed algorithm is compared with previous methods in term of the reconstruction accuracy and computation complexity. Numerous experiments demonstrate the superior performance of the proposed algorithm for compressed MR image reconstruction.

Chapter 6 introduces how to use the sparsity techniques to practical applications in image processing and computer-aided diagnosis. 1) Tag Separation in Cardiac Tagged MRI: sparsity
techniques are used to conduct tag separation in tMRIs for better cardiac boundary segmentation and tag tracking. While previous methods have focused on tag removal, our approach can acquire both optimally-decomposed tag-only image and the cardiac image without tags simultaneously, which can be used for better tag tracking and cardiac boundary segmentation. We demonstrate the superior performance of the proposed approach through extensive experiments on large sets of cardiac tagged MR images. 2) Cervigram Image Segmentation: we use the reconstructive sparse representation method to segment tissues in optical images of the uterine cervix. Because of large variations in image appearance, the color and texture features in optical images often overlap with each other and are not linearly separable. By leveraging sparse representations the data can be transformed to higher dimensions with sparse constraints and become more separated. Compared with other general methods, our approach showed lower space and time complexity and higher sensitivity. 3) Transformation-invariant sparse representation: Traditional sparse representation is sensitive to image plane transformations such that existing approaches can not reconstruct the sparse representation of a geometrically transformed image. A simple technique is proposed to obtain transformation-invariant image sparse representation. We then can simultaneously recover the sparse representation of a target image and the image plane transformation between the target and the model images. It has been successfully used in face recognition and dynamic texture registration and obtained improved performance over previous methods in these applications.

Chapter 7 concludes this thesis by surveying the contributions of my thesis work, and outline directions for future research. Many of these ideas integrate the aspects of data structures and sparsity priors guided by the proposed structured sparsity framework, which have complementary strengths.
Chapter 2
The Benefit of Group Sparsity

This chapter investigates the benefits of the sparsity with group structure. A theory is developed for Group Lasso using a concept called strong group sparsity. We theoretically prove that group Lasso is superior to standard Lasso for strongly group-sparse data.

2.1 Introduction

We are interested in the sparse learning problem for least squares regression. Consider a set of $p$ basis vectors $\{x_1, \ldots, x_p\}$ where $x_j \in \mathbb{R}^n$ for each $j$. Here, $n$ is the sample size.

Denote by $X$ the $n \times p$ data matrix, with column $j$ of $X$ being $x_j$. Given an observation $y = [y_1, \ldots, y_n] \in \mathbb{R}^n$ that is generated from a sparse linear combination of the basis vectors plus a stochastic noise vector $\epsilon \in \mathbb{R}^n$:

$$y = X\beta + \epsilon = \sum_{j=1}^{d} \beta_j x_j + \epsilon,$$

where we assume that the target coefficient $\beta$ is sparse. Throughout the chapter, we consider fixed design only. That is, we assume $X$ is fixed, and randomization is with respect to the noise $\epsilon$. Note that we do not assume that the noise $\epsilon$ is zero-mean.

Define the support of a sparse vector $\beta \in \mathbb{R}^p$ as

$$\text{supp}(\beta) = \{j : \beta_j \neq 0\},$$

and $\|\beta\|_0 = |\text{supp}(\beta)|$. A natural method for sparse learning is $L_0$ regularization:

$$\hat{\beta}_{L_0} = \arg \min_{\beta \in \mathbb{R}^p} \|X\beta - y\|^2_2 \quad \text{subject to} \quad \|\beta\|_0 \leq k,$$

where $k$ is the sparsity. Since this optimization problem is generally NP-hard, in practice, one often considers the following $L_1$ regularization problem, which is the closest convex relaxation of $L_0$:

$$\hat{\beta}_{L_1} = \arg \min_{\beta \in \mathbb{R}^p} \left[ \frac{1}{n} \|X\beta - y\|^2_2 + \lambda \|\beta\|_1 \right],$$
where $\lambda$ is an appropriately chosen regularization parameter and $n$ is the dimension of $y$. This method is often referred to as Lasso in the statistical literature.

In practical applications, one often knows a group structure on the coefficient vector $\beta$ so that variables in the same group tend to be zeros or nonzeros simultaneously. The purpose of this chapter is to show that if such a structure exists, then better results can be obtained.

### 2.2 Strong Group Sparsity

For simplicity, we shall only consider non-overlapping groups here, although our analysis can be adapted to handle moderately overlapping groups.

Assume that $\{1, \ldots, p\} = \bigcup_{j=1}^{m} G_j$ is partitioned into $m$ disjoint groups $G_1, G_2, \ldots, G_m$:

$L_i \cap G_j = \emptyset$ when $i \neq j$. Moreover, throughout the chapter, we let $k_j = |G_j|$, and $k_0 = \max_{j \in \{1, \ldots, m\}} k_j$. Given $S \subset \{1, \ldots, m\}$ that denotes a set of groups, we define $G_S = \bigcup_{j \in S} G_j$.

Given a subset of variables $F \subset \{1, \ldots, p\}$ and a coefficient vector $\beta \in \mathbb{R}^p$, let $\beta_F$ be the vector in $\mathbb{R}^{|F|}$ which is identical to $\beta$ in $F$. Similar, $X_F$ is the $n \times |F|$ matrix with columns identical to $X$ in $F$.

The following method, often referred to as group Lasso, has been proposed to take advantage of the group structure:

$$\hat{\beta} = \arg \min_{\beta} \left[ \frac{1}{n} \|X\beta - y\|_2^2 + \lambda \sum_{j=1}^{m} \|\beta_{G_j}\|_2 \right].$$

(2.1)

We are interested in developing a theory that characterizes the performance of (2.1). We are interested in conditions under which group Lasso yields better estimate of $\beta$ than the standard Lasso.

Instead of the standard sparsity assumption, where the complexity is measured by the number of nonzero coefficients $k$, we introduce the strong group sparsity concept below. The idea is to measure the complexity of a sparse signal using group sparsity in addition to coefficient sparsity.

**Definition 2.2.1** A coefficient vector $\beta \in \mathbb{R}^p$ is $(g, k)$ strongly group-sparse if there exists a set $S$ of groups such that

$$\text{supp}(\beta) \subset G_S, \quad |G_S| \leq k, \quad |S| \leq g.$$
does not work well when $k/\|\hat{\beta}\|_0$ is large. In that case, the signal is only weak group sparse, and one needs to use $\|\hat{\beta}\|_0$ to precisely measure the real sparsity of the signal. Unfortunately, such information is not included in the group Lasso formulation, and there is no simple fix of this problem using variations of group Lasso. This is because our theory requires that the group Lasso regularization term is strong enough to dominate the noise, and the strong regularization causes a bias of the order $O(k)$ which cannot be removed. This is one fundamental drawback which is inherent to the group Lasso formulation.

2.3 Related Work

The idea of using group structure to achieve better sparse recovery performance has received much attention. For example, group sparsity has been considered for simultaneous sparse approximation [140] and multi-task compressive sensing [72] from the Bayesian hierarchical modeling point of view. Under the Bayesian hierarchical model framework, data from all sources contribute to the estimation of hyper-parameters in the sparse prior model. The shared prior can then be inferred from multiple sources. Although the idea can be justified using standard Bayesian intuition, there are no theoretical results showing how much better (and under what conditions) the resulting algorithms perform.

In [121], the authors attempted to derive a bound on the number of samples needed to recover block sparse signals, where the coefficients in each block are either all zero or all nonzero. In our terminology, this corresponds to the case of group sparsity with equal size groups. The algorithm considered there is a special case of (2.1) with $\lambda_j \to 0^+$. However, their result is very loose, and does not demonstrate the advantage of group Lasso over standard Lasso.

In the statistical literature, the group Lasso (2.1) has been studied by a number of authors [148, 6, 96, 76, 101]. There were no theoretical results in [148]. Although some theoretical results were developed in [6, 96], neither showed that group Lasso is superior to the standard Lasso.

The authors of [76] showed that group Lasso can be superior to standard Lasso when each group is an infinite dimensional kernel, by using an argument completely different from ours (they relied on the fact that meaningful analysis can be obtained for kernel methods in infinite dimension). Their idea cannot be adapted to show the advantage of group Lasso in finite dimensional scenarios of interests such as in the standard compressive sensing setting. Therefore our analysis, which focuses on the latter, is complementary to their work.

The work most related to ours is [101], where the authors considered a special case of group
Lasso in the multi-task learning scenario, and showed that the number of samples required for recovering the exact support set is smaller for group Lasso under appropriate conditions. However, there are major differences between our analysis and their analysis. First, the group formulation we consider here is more general and includes the multi-task scenario as a special case. Second, we study signal recovery performance in 2-norm instead of the exact recovery of support set in their analysis. The sparse eigenvalue condition employed in this work is often considerably weaker than the irrepresentable type condition in their analysis (which is required for exact support set recovery). Third, our analysis also shows that for strongly group-sparse signals, even when the number of samples is large, the group Lasso can still have advantages in that it is more robust to noise than standard Lasso. Finally, the result obtained here, which relies on the strong group sparsity concept, is more general and intuitive.

In the above context, the main contribution of this work is the introduction of the strong group sparsity concept, under which a satisfactory theory of group Lasso is developed. Our result shows that strongly group sparse signals can be estimated more reliably using group Lasso, in that it requires fewer number of samples in the compressive sensing setting, and is more robust to noise in the statistical estimation setting.

2.4 Assumptions

The following assumption on the noise is important in our analysis. It captures an important advantage of group Lasso over standard Lasso under the strong group sparsity assumption.

**Assumption 2.4.1 (Group noise condition)** There exist non-negative constants $a, b$ such that for any fixed group $j \in \{1, \ldots, m\}$, and $\eta \in (0, 1)$: with probability larger than $1 - \eta$, the noise projection to the $j$-th group is bounded by:

$$\|((X_{G_j} \otimes X_{G_j})^{-0.5} X_{G_j}^\top (\epsilon - E\epsilon))\|_2 \leq a \sqrt{k_j} + b \sqrt{-\ln \eta}. $$

The importance of the assumption is that the concentration term $\sqrt{-\ln \eta}$ does not depend on $k$. This reveals a significant benefit of group Lasso over standard Lasso: that is, the concentration term does not increase when the group size increases. This implies that if we can correctly guess the group sparsity structure, the group Lasso estimator is more stable with respect to stochastic noise than the standard Lasso.

We shall point out that this assumption holds for independent sub-Gaussian noise vectors, where $e^{t(\epsilon_i - E\epsilon_i)} \leq e^{t^2 \sigma^2/2}$ for all $t$ and $i = 1, \ldots, n$. It can be shown that one may choose $a = 2.8$ and $b = 2.4$ when $\eta \in (0, 0.5)$. Since a complete treatment of sub-Gaussian noise is not
important for the purpose of this chapter, we only prove this assumption under independent Gaussian noise, which can be directly calculated.

**Proposition 2.4.1** Assume the noise vector $\epsilon$ are independent Gaussians: $\epsilon_i - \mathbb{E}\epsilon_i \sim N(0, \sigma^2)$, where each $\sigma_i \leq \sigma$ ($i = 1, \ldots, n$). Then Assumption 3.6.1 holds with $a = \sigma$ and $b = \sqrt{2}\sigma$.

The next assumption handles the case that true target is not exactly sparse. That is, we only assume that $X\beta \approx \mathbb{E}y$.

**Assumption 2.4.2 (Group approximation error condition)** There exist $\delta_a, \delta_b \geq 0$ such that for all group $j \in \{1, \ldots, m\}$: the projection of error mean $\mathbb{E}\epsilon$ to the $j$-th group is bounded by:

$$
\|(X_{G_j}^\top X_{G_j})^{-0.5} X_{G_j}^\top \mathbb{E}\epsilon\|_2/\sqrt{n} \leq \sqrt{k_j} \delta_a + \delta_b.
$$

As mentioned earlier, we do not assume that the noise is zero-mean. Hence $\mathbb{E}\epsilon$ may not equal zero. In other words, this condition considers the situation that the true target is not exactly sparse. It resembles algebraic noise in [154] but takes the group structure into account. Similar to [154], if we assume approximation error of the form

$$
\frac{1}{n} \|X\hat{\beta} - \mathbb{E}y\|_2^2 \leq \delta^2,
$$
then one can find a target $\hat{\beta}'$ with similar sparsity so that $\delta_a = O(\delta/\sqrt{k})$ and $\delta_b = 0$. In Theorem 2.5.1 below, the contribution of this approximation error to the reconstruction error $\|\hat{\beta} - \beta\|_2$ is $O(\delta)$. The detailed derivation of the estimate $\delta_a = O(\delta/\sqrt{k})$ and $\delta_b = 0$ will be skipped because they do not show the benefit of group Lasso over standard Lasso. Therefore in order to compare our results to that of the standard Lasso, one may consider the simple situation where $\delta_a = \delta_b = 0$. That is, the target is exactly sparse. The only reason to include Assumption 2.4.2 is to illustrate that our analysis can handle approximate sparsity.

The last assumption is a sparse eigenvalue condition, used in the modern analysis of Lasso (e.g., [16, 154]). It is also closely related to (and slightly weaker than) the RIP (restricted isometry property) assumption [22] in the compressive sensing literature. This assumption takes advantage of group structure, and can be considered as (a weaker version of) group RIP.

We introduce a definition before stating the assumption.

**Definition 2.4.1** For all $F \subset \{1, \ldots, p\}$, define

$$
\rho_-(F) = \inf \left\{ \frac{1}{n} \|X\beta\|_2^2/\|\beta\|_2^2 : \text{supp}(\beta) \subset F \right\},
$$

$$
\rho_+(F) = \sup \left\{ \frac{1}{n} \|X\beta\|_2^2/\|\beta\|_2^2 : \text{supp}(\beta) \subset F \right\}.
$$
Moreover, for all $1 \leq s \leq p$, define

$$
\rho_-(s) = \inf \{ \rho_-(G_S) : S \subset \{1, \ldots, m\}, |G_S| \leq s \},
$$

$$
\rho_+(s) = \sup \{ \rho_+(G_S) : S \subset \{1, \ldots, m\}, |G_S| \leq s \}.
$$

**Assumption 2.4.3 (Group sparse eigenvalue condition)** There exist $s, c > 0$ such that

$$
\frac{\rho_+(s) - \rho_-(2s)}{\rho_-(s)} \leq c.
$$

Assumption 2.4.3 illustrates another advantage of group Lasso over standard Lasso. Since we only consider eigenvalues for sub-matrices consistent with the group structure $\{G_j\}$, the ratio $\rho_+(s)/\rho_-(s)$ can be significantly smaller than the corresponding ratio for Lasso (which considers all subsets of $\{1, \ldots, p\}$ up to size $s$). For example, assume that all group sizes are identical $k_1 = \ldots = k_m = k_0$, and $s$ is a multiple of $k_0$. For random projections used in compressive sensing applications, only $n = O(s + (s/k_0) \ln m)$ projections are needed for Assumption 2.4.3 to hold. In comparison, for standard Lasso, we need $n = O(s \ln p)$ projections. The difference can be significant when $p$ and $k_0$ are large. More precisely, we have the following random projection sample complexity bound for the group sparse eigenvalue condition. Although we assume Gaussian random matrix in order to state explicit constants, it is clear that similar results hold for other sub-Gaussian random matrices.

**Proposition 2.4.2 (Group-RIP)** Suppose that elements in $X$ are iid standard Gaussian random variables $N(0, 1)$. For any $t > 0$ and $\delta \in (0, 1)$, let

$$
n \geq \frac{8}{\delta^2} \left[ \ln 3 + t + k \ln(1 + 8/\delta) + g \ln(em/g) \right].
$$

Then with probability at least $1 - e^{-t}$, the random matrix $X \in \mathbb{R}^{n \times p}$ satisfies the following group-RIP inequality for all $(g, k)$ strongly group-sparse vector $\tilde{\beta} \in \mathbb{R}^p$,

$$
(1 - \delta)\|\tilde{\beta}\|_2 \leq \frac{1}{\sqrt{n}}\|X\tilde{\beta}\|_2 \leq (1 + \delta)\|\tilde{\beta}\|_2.
$$

(2.2)

### 2.5 Main Results

Our main result is the following signal recovery (2-norm parameter estimation error) bound for group Lasso.

**Theorem 2.5.1** Suppose that Assumption 3.6.1, Assumption 2.4.2, and Assumption 2.4.3 are valid. Take $\lambda_j = (A\sqrt{k_j} + B)/\sqrt{n}$, where both $A$ and $B$ can depend on data $y$. Given $\eta \in (0, 1)$, with probability larger than $1 - \eta$, if the following conditions hold:

...
• $A \geq 4 \max_j \rho_+(G_j)^{1/2}(a + \delta a \sqrt{n})$,

• $B \geq 4 \max_j \rho_+(G_j)^{1/2}(b \sqrt{\ln(m/\eta)} + \delta b \sqrt{n})$,

• $\hat{\beta}$ is a $(g, k)$ strongly group-sparse coefficient vector,

• $s \geq k + k_0$,

• Let $\ell = s - (k - k_0) + 1$, and $g_\ell = \min\{|S| : |G_S| \geq \ell, S \subset \{1, \ldots, m\}$, we have

$$c^2 \leq \frac{\ell A^2 + g_\ell B^2}{72(kA^2 + gB^2)},$$

then the solution of (2.1) satisfies:

$$\|\hat{\beta} - \bar{\beta}\|_2 \leq \frac{\sqrt{4.5}}{p_- (s) \sqrt{n}} (1 + 0.25c^{-1}) \sqrt{A^2 k + gB^2}.$$

The first four conditions of the theorem are not critical, as they are just definitions and choices for $\lambda_j$. The fifth assumption is critical, which means that the group sparse eigenvalue condition has to be satisfied with some $c$ that is not too large. In order to satisfy the condition, $\ell$ should be chosen relatively large as the right hand side is linear in $\ell$. However, this implies that $s$ also grow linearly. It is possible to find $s$ so that the condition is satisfied when $c^2$ in Assumption 2.4.3 grows sub-linearly in $s$. Consider the situation that $\delta a = \delta b = 0$. If the conditions of Theorem 2.5.1 is satisfied, then

$$\|\hat{\beta} - \bar{\beta}\|_2 = O((k + g \ln(m/\eta))/n).$$

In comparison, The Lasso estimator can only achieve the bound

$$\|\hat{\beta}_{L1} - \bar{\beta}\|_2 = O((\|\bar{\beta}\|_0 \ln(p/\eta))/n).$$

If $k/\|\bar{\beta}\|_0 \ll \ln(p/\eta)$ (which means that the group structure is useful) and $g \ll \|\bar{\beta}\|_0$, then the group Lasso is superior. This is consistent with intuition. However, if $k \gg \|\bar{\beta}\|_0 \ln(p/\eta)$, then group Lasso is inferior. This happens when the signal is not strongly group sparse.

Theorem 2.5.1 also suggests that if the group sizes are not even, then group Lasso may not work well when the signal is contained in small sized groups. This is because in such case $g_\ell$ can be significantly smaller than $g$ even with relatively large $\ell$, which means we have to choose a large $s$ and small $c$, implying a poor bound. This prediction is confirmed in Section 2.6.2 using simulated data. Intuitively, group Lasso favors large sized groups because the 2-norm regularization for large group size is weaker. Adjusting regularization parameters $\lambda_j$ not only fails to work in theory, but also impractical since it is unrealistic to tune many parameters. This
Suppose that Assumption 3.6.1 and Assumption 2.4.2 are valid with \( a = b = \delta b = 0 \). Take \( \lambda_j = 4\sqrt{k_j} \max_j \rho_+(G_j)^{1/2} \delta a \). Let \( \hat{\beta} \) be a \((k, g)\) strongly group-sparse signal, \( \ell = k \), and \( s = 2k + k_0 - 1 \). If \((\rho_+(s) - \rho_-(2s))/\rho_-(s) \leq 1/\sqrt{72} \), then the solution of (2.1) satisfies:

\[
\|\hat{\beta} - \beta\|_2 \leq \frac{6\sqrt{2} + 18}{\rho_-(s)} \max_j \rho_+(G_j)^{1/2} \delta a \sqrt{k}.
\]

If \( \delta a = 0 \), then we can achieve exact recovery. Moreover, as pointed out below Assumption 2.4.2, \( \delta a \sqrt{k} = O(\|X\hat{\beta} - y\|_2^2/n) \). A more detailed explanation can be found in [154]. If we have even sized groups, the number of samples \( n \) required for Corollary 2.5.1 to hold is \( O(k + g \ln(m/g)) \), where \( g = k/k_0 \). In comparison, although a similar result holds for Lasso, it requires sample size of order \( \|\hat{\beta}\|_0 \ln(p/\|\hat{\beta}\|_0) \). Again, group Lasso has a significant advantage if \( k/\|\hat{\beta}\|_0 \ll \ln(p/\|\hat{\beta}\|_0) \), \( g \ll \|\hat{\beta}\|_0 \), and \( p \) is large.

The following corollary is for even sized groups, and the result is simpler to interpret. For standard Lasso, \( B = O(\sqrt{m}p) \), and for group Lasso, \( B = O(\sqrt{\ln m}) \). The benefit of group Lasso is the division of \( B^2 \) by \( k_0 \) in the bound, which is a significant improvement when the dimensionality \( p \) is large. The disadvantage of group Lasso is that the signal sparsity \( \|\hat{\beta}\|_0 \) is replaced by the group sparsity \( k \). This is not an artifact of our analysis, but rather a fundamental drawback inherent to the group Lasso formulation. The effect is observable, as shown in our simulation studies.

**Corollary 2.5.2 (Even group size)** Suppose that Assumption 3.6.1 and Assumption 2.4.2 are valid. Assume also that all groups are of equal sizes: \( k_0 = k_j \) for \( j = 1, \ldots, m \). Given \( \eta \in (0, 1) \), let

\[
\lambda_j = (A\sqrt{k_0} + B)/\sqrt{\eta},
\]

where \( A \geq 4 \max_j \rho_+(G_j)^{1/2} (a + \delta a \sqrt{\eta}) \) and \( B \geq 4 \max_j \rho_+(G_j)^{1/2} (b\sqrt{\ln(m/\eta)} + \delta b \sqrt{\eta}) \). Let \( \hat{\beta} \) be a \((k, k/k_0)\) strongly group-sparse signal. With probability larger than \( 1 - \eta \), if

\[
6\sqrt{2}(\rho_+(k + \ell) - \rho_-(2k + 2\ell))/\rho_-(k + \ell) < \sqrt{\ell/k}
\]

for some \( \ell > 0 \) that is a multiple of \( k_0 \), then the solution of (2.1) satisfies:

\[
\|\hat{\beta} - \beta\|_2 \leq \rho_-(k + \ell)^{-1}(\sqrt{4.5} + 4.5\ell/k)\sqrt{A^2 + B^2/k_0 \sqrt{k/\eta}}.
\]
2.6 Experiments

We want to verify our theory by comparing group Lasso to Lasso on simulation data. For quantitative evaluation, the recovery error is defined as the relative difference in 2-norm between the estimated sparse coefficient vector $\hat{\beta}$ and the ground-truth sparse coefficient $\bar{\beta}$:

$$\frac{\|\hat{\beta} - \bar{\beta}\|_2}{\|\bar{\beta}\|_2}.$$ 

The regularization parameter $\lambda$ in Lasso is chosen with five-fold cross validation. In group Lasso, we simply set the regularization parameter $\lambda_j = (\lambda \sqrt{k_j})/\sqrt{n}$ for $j = 1, 2, ..., m$. The regularization parameter $\lambda$ is then chosen with five-fold cross validation. Here we set $B = 0$ in the formula $\lambda_j = O(A \sqrt{k_j} + B)$. Since the relative performance of group Lasso versus standard Lasso is similar with other values of $B$, in order to avoid redundancy, we do not include results with $B \neq 0$.

2.6.1 Even group size

In this set of experiments, the projection matrix $X$ is generated by creating an $n \times p$ matrix with i.i.d. draws from a standard Gaussian distribution $N(0, 1)$. For simplicity, the rows of $X$ are normalized to unit magnitude. Zero-mean Gaussian noise with standard deviation $\sigma = 0.01$ is added to the measurements. Our task is to compare the recovery performance of Lasso and Group Lasso for these $(g, k)$ strongly group sparse signals.

With correct group structure

In this experiment, we randomly generate $(g, k)$ strongly group sparse coefficients with values $\pm 1$, where $p = 512$, $k = 64$ and $g = 16$. There are 128 groups with even group size of $k_0 = 4$. Here the group structure coincides with the signal sparsity: $k = \|\bar{\beta}\|_0$.

Figure 2.1 shows an instance of generated sparse coefficient vector and the recovered results by Lasso and group Lasso respectively when $n = 3k = 192$. Since the sample size $n$ is only three times the signal sparsity $k$, the standard Lasso does not achieve good recovery results, whereas the group Lasso achieves near perfect recovery of the original signal.

Figure 2.2(a) shows the effect of sample size $n$, where we report the average recover error over 100 random runs for each sample size. Group Lasso is clearly superior in this case. These results show that the group Lasso can achieve better recovery performance for $(g, k)$ strongly group sparse signals with fewer measurements, which is consistent with our theory.

To study the effect of the group number $g$ (with $k$ fixed), we set the sample size $n = 160$ and then change the group number while fixing other parameters. Figure 2.2(b) shows the
Figure 2.1: Recovery results when the assumed group structure is correct. (a) Original data; (b) results with Lasso (recovery error is 0.3444); (c) results with Group Lasso (recovery error is 0.0419).

Figure 2.2: Recovery performance: (a) recovery error vs. sample size ratio $n/k$; (b) recovery error vs. group number $g$. 
recovery performance of the two algorithms, averaged over 100 random runs for each sample size. As expected, the recovery performance for Lasso is independent to the group number within statistical error. Moreover, the recovery results for group Lasso are significantly better when the group number $g$ is much smaller than the sparsity $k = 64$. When $g = k$, the group Lasso becomes identical to Lasso, which is expected. This shows that the recovery performance of group Lasso degrades when $g/k$ increases, which confirms our theory.

**With incorrect group structure**

In this experiment, we assume that the known group structure is not exactly the same as the sparsity of the signal (that is, $k > \|\hat{\beta}\|_0$). We randomly generate strongly group sparse coefficients with values $\pm 1$, where $p = 512$, $\|\hat{\beta}\|_0 = 64$ and $g = 16$. In the first experiment, we let $k = 4\|\hat{\beta}\|_0$, and use $m = 32$ groups with even group size of $k_0 = 16$.

Figure 2.3 shows one instance of the generated sparse signal and the recovered results by Lasso and group Lasso respectively when $n = 3\|\hat{\beta}\|_0 = 192$. In this case, the standard Lasso obtains better recovery results than the group Lasso. Figure 2.2(a) shows the effect of sample size $n$, where we report the averaged recover error over 100 random runs for each sample size. The group Lasso recovery performance is clearly inferior to that of the Lasso. This shows that group Lasso fails when $k/\|\hat{\beta}\|_0$ is relatively large, which is consistent with our theory.

To study the effect of $k/\|\hat{\beta}\|_0$ on the group Lasso performance, we keep $\|\hat{\beta}\|_0$ fixed, and simply vary the group size as $k_0 = 1, 2, 4, 8, 16, 32, 64$ with $k/\|\hat{\beta}\|_0 = 1, 1, 1, 2, 4, 8, 16$. Figure 2.4(b) shows the performance of the two algorithms with different group sizes $k_0$ in terms of recovery error. It shows that the performance of group Lasso is better when $k/\|\hat{\beta}\|_0 = 1$. However, when $k/\|\hat{\beta}\|_0 > 1$, the performance of group Lasso deteriorates.

**2.6.2 Uneven group size**

In this set of experiments, we randomly generate $(g, k)$ strongly sparse coefficients with values $\pm 1$, where $p = 512$, and $g = 4$. There are 64 uneven sized groups. The projection matrix $X$ and noises are generated as in the even group size case. Our task is to compare the recovery performance of Lasso and Group Lasso for $(g, k)$ strongly sparse signals with $\|\hat{\beta}\|_0 = k$. To reduce the variance, we run each experiment 100 times and report the average performance.

In the first experiment, the group sizes of 64 groups are randomly generated and the $g = 4$ active groups are randomly extracted from these 64 groups. Figure 2.5(a) shows the recovery performance of Lasso and group Lasso with increasing sample size (measurements) in terms of
Figure 2.3: Recovery results when the assumed group structure is incorrect. (a) Original data; (b) results with Lasso (recovery error is 0.3616); (c) results with Group Lasso (recovery error is 0.6688).

Figure 2.4: Recovery performance: (a) recovery error vs. sample size ratio $n/k$; (b) recovery error vs. group size $k_0$. 
recovery error. Similar to the case of even group size, the group Lasso obtains better recovery results than those with Lasso. It shows that the group Lasso is superior when the group sizes are randomly uneven.

Figure 2.5: Recovery performance: (a) $g$ active groups have randomly uneven group sizes; (b) half of $g$ active groups are single element groups and another half of $g$ active groups have large group size.

As discussed after Theorem 2.5.1, because group Lasso favors large sized groups, if the signal is contained in small sized groups, then the performance of group Lasso can be relatively poor. In order to confirm this claim of Theorem 2.5.1, we consider the special case where 32 groups have large group sizes and each of the remaining 32 groups has only one element. First, we consider the case where half of $g = 4$ active groups are extracted from the single element groups and the other half of $g = 4$ active groups are extracted from the groups with large size. Figure 2.5(b) shows the signal recovery performance of Lasso and group Lasso. It is clear that the group Lasso performs better, but the results are not as good as those of Figure 2.5(a).

Moreover, Figure 2.6(a) shows the recovery performance of Lasso and group Lasso when all of the $g = 4$ active groups are extracted from large sized groups. We observe that the relative performance of group Lasso improves. Finally, Figure 2.6(b) shows the recovery performance of Lasso and group Lasso when all of the $g = 4$ active groups are extracted from single element groups. It is obvious that the group Lasso is inferior to Lasso in this case. This confirms the prediction of Theorem 2.5.1 that suggests that group Lasso favors large sized groups.

### 2.7 Discussions

This chapter introduces a concept called strong group sparsity that characterizes the signal recovery performance of group Lasso. In particular, we show that group Lasso is superior to
standard Lasso when the underlying signal is strongly group-sparse:

- Group Lasso is more robust to noise due to the stability associated with group structure.
- Group Lasso requires a smaller sample size to satisfy the sparse eigenvalue condition required in the modern sparsity analysis.

However, group Lasso can be inferior if the signal is only weakly group-sparse, or covered by groups with small sizes. Moreover, group Lasso does not perform well with overlapping groups (which is not analyzed in this chapter). Better learning algorithms may be needed to overcome these limitations.

### 2.8 Proofs

#### 2.8.1 Proof of Proposition 2.4.1

Without loss of generality, we may assume $\sigma_i > 0$ for all $i$ (otherwise, we can still let $\sigma_i > 0$ and then just take the limit $\sigma_i \to 0$ for some $i$).

For notation simplicity, we remove the subscript $j$ from the group index, and consider group $G$ with $k$ variables.

Let $\Sigma$ be the diagonal matrix with $\sigma_i$ as its diagonal elements. We can find an $n \times k$ matrix $Z = X_G (X_G^\top \Sigma X_G)^{-0.5}$, such that $Z^\top \Sigma Z = I_{k \times k}$. Let $\xi = Z^\top (\epsilon - E\epsilon) \in \mathbb{R}^k$. Since $\forall v \in \mathbb{R}^n$,

$$\| (X_G^\top X_G)^{-0.5} X_G^\top v \|_2 = \| (Z^\top Z)^{-0.5} Z^\top v \|_2,$$
we have

\[
\begin{align*}
\frac{\|X_G^\top X_G\|_2^{-0.5} X_G^\top (\epsilon - \mathbb{E}\epsilon)\|_2^2}{\xi^\top \xi} &\leq \sup_{v \in \mathbb{R}^n} \frac{v^\top Z (Z^\top Z)^{-1} Z^\top v}{v^\top ZZ^\top v} \\
&= \sup_{u \in \mathbb{R}^k} \frac{u^\top (Z^\top Z)^{-1} u}{u^\top u} = \sup_{u \in \mathbb{R}^k} \frac{u^\top Z^\top \Sigma Zu}{u^\top (Z^\top Z)u} \\
&\leq \sup_{v \in \mathbb{R}^n} \frac{v^\top \Sigma v}{v^\top v} \leq \sigma^2.
\end{align*}
\]

Therefore, we only need to show that with probability at least \(1 - \eta\) for all \(\eta \in (0, 1)\):

\[
\|\xi\|_2 \leq a\sqrt{k} + b\sqrt{-\ln \eta}
\]

with \(a = 1\) and \(b = \sqrt{2}\).

To prove this inequality, we note that the condition \(Z^\top \Sigma Z = I_{k \times k}\) means that the covariance matrix of \(\xi\) is \(I_{k \times k}\). Therefore the components of \(\xi\) are \(k\) iid Gaussians \(N(0, 1)\), and the distribution of \(\|\xi\|_2^2\) is \(\chi^2\). Many methods have been suggested to approximate the tail probability of \(\chi^2\) distribution. For example, a well-known approximation of \(\|\xi\|_2^2\) is the normal \(N(\sqrt{k} - 0.5, 0.5)\), which would imply \(a = b = 1\) in (2.3). In the following, we derive a slightly weaker tail probability bound using direct integration of tail probability for \(\delta \geq \sqrt{k}\):

\[
P(\|\xi\|_2^2 \geq \delta^2) = \frac{1}{\Gamma(k/2)2^{k/2}} \int_{x \geq \delta} x^{k/2 - 1}e^{-x/2}dx = \frac{2}{\Gamma(k/2)2^{k/2}} \int_{x \geq \delta} x^{k-1}e^{-x^2/2}dx = \frac{2\delta^{k-1}}{\Gamma(k/2)2^{k/2}} \int_{x \geq 0} e^{-(x+\delta)^2/2+(k-1)}\ln(1+x/\delta)dx \leq \frac{2\delta^{k-1}e^{-\delta^2/2}}{\Gamma(k/2)2^{k/2}} \int_{x \geq 0} e^{-x^2/2+\delta+(k-1)/\delta}dx \leq \frac{\sqrt{2}\pi\delta^{k-1}e^{-\delta^2/2}}{\Gamma(k/2)2^{k/2}} \leq \frac{\sqrt{0.5}(\delta/\sqrt{k})^{k-1}e^{-0.5\delta^2+0.5k}}{\Gamma(0.5k)0.5k} \leq \sqrt{0.5}e^{-\delta^2/2+0.5k+(k-1)(\delta/\sqrt{k})^2} \leq \sqrt{0.5}e^{-(\delta^2/2-\delta/\sqrt{k})^2/2}.
\]

This implies that (2.3) holds with \(a = 1\) and \(b = \sqrt{2}\).

Note that in the above derivation, we have used the following Sterling lower bound for the Gamma function

\[
\Gamma(0.5k) \geq \sqrt{2\pi(0.5k)^{0.5k-0.5}e^{-0.5k}}.
\]

### 2.8.2 Proof of Proposition 2.4.2

The following lemma is taken from [104]. Since the proof is simple, it is included for completeness.
Lemma 2.8.1 Consider the unit sphere $S^{k-1} = \{ x : \|x\|_2 = 1 \}$ in $\mathbb{R}^k$ ($k \geq 1$). Given any $\varepsilon > 0$, there exists an $\varepsilon$-cover $Q \subset S^{k-1}$ such that $\min_{q \in Q} \|x - q\|_2 \leq \varepsilon$ for all $\|x\|_2 = 1$, with $|Q| \leq (1 + 2/\varepsilon)^k$.

Proof Let $B^k = \{ x : \|x\|_2 \leq 1 \}$ be the unit ball in $\mathbb{R}^k$. Let $Q = \{ q_i \}_{i=1}^{\lfloor |Q| \rfloor} \subset S^{k-1}$ be a maximal subset such that $\|q_i - q_j\|_2 > \varepsilon$ for all $i \neq j$. By maximality, $Q$ is an $\varepsilon$-cover of $S^{k-1}$.

Since the balls $q_i + (\varepsilon/2)B^k$ are disjoint and belong to $(1 + \varepsilon/2)B^k$, we have

$$\sum_{i \leq |Q|} \text{vol}(q_i + (\varepsilon/2)B^k) \leq \text{vol}((1 + \varepsilon/2)B^k).$$

Therefore,

$$|Q|(\varepsilon/2)^k \text{vol}(B^k) \leq (1 + \varepsilon/2)^k \text{vol}(B^k),$$

which implies that $|Q| \leq (1 + 2/\varepsilon)^k$. \hfill \qed

The following concentration result for $\chi^2$ distribution is similar to Proposition 2.4.1. This is where the Gaussian assumption is used in the proof. A similar result holds for sub-Gaussian random variables.

Lemma 2.8.2 Let $\xi \in \mathbb{R}^n$ be a vector of $n$ iid standard Gaussian variables: $\xi_i \sim N(0,1)$. Then $\forall \varepsilon > 0$:

$$\Pr \left[ \|\xi\|_2 - \sqrt{n} \geq \varepsilon \right] \leq 3e^{-\varepsilon^2/2}. $$

Proof Proposition 2.4.1 implies that

$$\Pr \left[ \|\xi\|_2 - \sqrt{n} \geq \varepsilon \right] \leq \sqrt{0.5}e^{-\varepsilon^2/2}. $$

Using identical derivation in the proof of Proposition 2.4.1, and let $\delta = \sqrt{n} - \varepsilon$ and $k = n$, we obtain:

$$\Pr \left[ \|\xi\|_2 - \sqrt{n} \leq -\varepsilon \right] \leq \frac{2\delta^{k-1}e^{-\delta^2/2}}{\Gamma(k/2)2^{k/2}} \int_{x \leq 0} e^{-x^2/2 + x(-\delta + (k-1)/\delta)} dx$$

$$\leq \frac{2\delta^{k-1}e^{-\delta^2/2}}{\Gamma(k/2)2^{k/2}} \int_{x \leq 0} e^{-x^2/2 - x} dx$$

$$\leq 3 \times \frac{\sqrt{2\pi} \delta^{k-1}e^{-\delta^2/2}}{\Gamma(k/2)2^{k/2}} \leq 3 \times \sqrt{0.5}e^{-\varepsilon^2/2}. $$

Combining the above two inequalities, we obtain the desired bound. \hfill \qed

The derivation of the following estimate employs a standard proof technique (for example, see [?]).
Lemma 2.8.3 Suppose $X$ is generated according to Proposition 2.4.2. For any fixed set $S \subset \{1, \ldots, p\}$ with $|S| = k$ and $0 < \delta < 1$, we have with probability exceeding $1 - 3(1 + 8/\delta)^k e^{-n\delta^2/8}$:

\[
(1 - \delta) \|\beta\|_2 \leq \frac{1}{\sqrt{n}} \|X_S\beta\|_2 \leq (1 + \delta) \|\beta\|_2 \tag{2.4}
\]

for all $\beta \in \mathbb{R}^k$.

**Proof** It is enough to prove the conclusion in the case of $\|\beta\|_2 = 1$. According to Lemma 2.8.1, given $\epsilon_1 > 0$, there exists a finite set $Q = \{q_i\}$ with $|Q| \leq (1 + 2/\epsilon_1)^k$ such that $\|q_i\|_2 = 1$ for all $i$, and $\min_i \|\beta - q_i\|_2 \leq \epsilon_1$ for all $\|\beta\|_2 = 1$.

For each $i$, Since elements of $\xi = X_S q_i$ are iid Gaussians $\mathcal{N}(0, 1)$, Lemma 2.8.2 implies that $\forall \epsilon_2 > 0$:

\[
\Pr \left[ \|X_S q_i\|_2 - \sqrt{n}\|q_i\|_2 \geq \sqrt{n}\epsilon_2 \right] \leq 3e^{-n\epsilon_2^2/2}.
\]

Taking union bound for all $q_i \in Q$, we obtain with probability exceeding $1 - 3(1 + 2/\epsilon_1)^k e^{-\epsilon_2^2/2}$:

\[
(1 - \epsilon_2) \leq \frac{1}{\sqrt{n}} \|X_S q_i\|_2 \leq (1 + \epsilon_2).
\]

Now, we define $\rho$ as the smallest nonnegative number such that

\[
\frac{1}{\sqrt{n}} \|X_S\beta\|_2 \leq (1 + \rho) \tag{2.5}
\]

for all $\beta \in \mathbb{R}^k$ with $\|\beta\|_2 = 1$. Since for all $\|\beta\|_2 = 1$, we can find $q_i \in Q$ such that $\|\beta - q_i\|_2 \leq \epsilon_1$, we have

\[
\|X_S\beta\|_2 \leq \|X_S q_i\|_2 + \|X_S(\beta - q_i)\|_2 \leq \sqrt{n}(1 + \epsilon_2 + (1 + \rho)\epsilon_1),
\]

where we used (2.5) in the derivation. Since $\rho$ is the smallest non-negative constant for which (2.5) holds, we have

\[
\sqrt{n}(1 + \rho) \leq \sqrt{n}(1 + \epsilon_2 + (1 + \rho)\epsilon_1),
\]

which implies that

\[
\rho \leq (\epsilon_1 + \epsilon_2)/(1 - \epsilon_1).
\]

Now we choose $\epsilon_1 = \delta/4$ and $\epsilon_2 = \delta/2$. Since $0 < \delta < 1$, it is easy to see that $\rho \leq \delta$. This proves the upper bound. For the lower bound, we note that for all $\|\beta\|_2 = 1$ with $\|\beta - q_i\|_2 \leq \epsilon_1$, we have

\[
\|X_S\beta\|_2 \geq \|X_S q_i\|_2 - \|X_S(\beta - q_i)\|_2 \geq \sqrt{n}(1 - \epsilon_2 - (1 + \rho)\epsilon_1),
\]

which leads to the desired result. \hfill \qed
Proof of Proposition 2.4.2

For each subset \( S \subset \{1, \ldots, m\} \) of groups with \( |S| \leq g \) and \( |G_S| \leq k \), we know from 3.9.3 that for all \( \beta \) such that \( \text{supp}(\beta) \subset G_S \):

\[
(1 - \delta)\|\beta\|_2 \leq \|X\beta\|_2 \leq (1 + \delta)\|\beta\|_2
\]

with probability exceeding \( 1 - 3(1 + 8/\delta)^k e^{-n\delta^2/8} \).

Since the number of such groups \( S \) can be no more than \( C g^m \leq \left(\frac{em}{g}\right)^g \), by taking the union bound, we know that the group RIP in Equation (3.3) fails with probability less than

\[
3(\frac{em}{g})^g(1 + 8/\delta)^k e^{-n\delta^2/8} \leq e^{-t}.
\]

2.8.3 Technical Lemmas

The following lemmas are adapted from [154] to handle group sparsity structure. Similar techniques can be found in [16]. The first lemma is in [154]. The proof is included for completeness.

**Lemma 2.8.4** Let \( A = X^\top X/n \), and let \( I \) and \( J \) be non-overlapping indices in \( \{1, \ldots, p\} \). We have

\[
\|A_{I,J}\|_2 \leq \sqrt{(\rho_+(I) - \rho_-(I \cup J))(\rho_+(J) - \rho_-(I \cup J))},
\]

where the matrix 2-norm is defined as \( \|A_{I,J}\|_2 = \sup_{\|u\|_2 = \|v\|_2 = 1} |u^\top A_{I,J}v| \).

**Proof** Consider \( v \in \mathbb{R}^p \) with \( v_I \in \mathbb{R}^{\|I\|} \) and \( v_J \in \mathbb{R}^{\|J\|} \); positive semi-definiteness implies that

\[
\begin{align*}
\rho_+(I)\|v_I\|_2^2 &+ 2tv_I^\top A_{I,J}v_J + t^2\rho_+(J)\|v_J\|_2^2 \\
\geq &v_I^\top A_{I,I}v_I + 2tv_I^\top A_{I,J}v_J + t^2v_J^\top A_{J,J}v_J \\
\geq &\rho_-(I \cup J)(\|v_I\|_2^2 + t^2\|v_J\|_2^2)
\end{align*}
\]

for all \( t \). This implies that

\[
|v_I^\top A_{I,J}v_J| \leq \sqrt{(\rho_+(I) - \rho_-(I \cup J))(\rho_+(J) - \rho_-(I \cup J))}\|v_I\|_2\|v_J\|_2,
\]

which leads to the desired result. \( \blacksquare \)

The next lemma uses the previous result to control the contribution of the non-signal part \( G^c \) of the error vector \( u \) (in terms of its group \( L_1 \) norm) to the quantity \( u_{G^c}^\top A_{G,G^c}u_{G^c} \).

**Lemma 2.8.5** Given \( u \in \mathbb{R}^p \) and \( S \subset \{1, \ldots, m\} \). Consider \( \ell \geq 1 \) and define

\[
\lambda_\ell^2 = \min \left\{ \sum_{j \in S'} \lambda_j^2 : |G_{S'}| \geq \ell \right\}.
\]

\[
\lambda_\ell^2 = \min \left\{ \sum_{j \in S'} \lambda_j^2 : |G_{S'}| \geq \ell \right\}.
\]
Let $S_0 \subset \{1, \ldots, m\} - S$ contain indices $j$ of largest values of $\|u_{G_j}\|_2/\lambda_j$ ($j \notin S$), and satisfies the condition $\ell \leq |G_{S_0}| < \ell + k_0$. Let $G = G_S \cup G_{S_0}$. Then

$$\sqrt{\sum_{j \notin S \cup S_0} \|u_{G_j}\|_2^2} \leq (2\lambda_-)^{-1} \sum_{j \notin S} \lambda_j \|u_{G_j}\|_2$$

and

$$\frac{1}{n} \sum_{j \notin S \cup S_0} u_j^T X_G^T X_G u_{G_j} \leq \lambda_-^{-1} \tilde{\rho}_+ \|u_G\|_2 \sum_{j \notin S} \lambda_j \|u_{G_j}\|_2,$$

where $\tilde{\rho}_+ = \sqrt{(\rho_+(G) - \rho_-(|G| + \ell + k_0 - 1)) (\rho_+(\ell + k_0 - 1) - \rho_-(|G| + \ell + k_0 - 1))}$.

**Proof** Without loss of generality, we assume that $S = \{1, \ldots, g\}$, and we assume that $j > g$ is in descending order of $\|u_{G_j}\|_2/\lambda_j$. Let $S_0, S_1, \ldots$ be the first, second, etc, consecutive blocks of $j > g$, such that $\ell \leq |G_{S_k}| < \ell + k_0$ (except for the last $S_k$). If we let $G^k = G_{S_k}$, then:

$$\sum_{j \notin S \cup S_0} \|u_{G_j}\|_2^2 \leq \sum_{j \notin S \cup S_0} \lambda_j \|u_{G_j}\|_2 \left[ \max_{j \notin S \cup S_0} \|u_{G_j}\|_2/\lambda_j \right]$$

$$\leq \sum_{j \notin S \cup S_0} \lambda_j \|u_{G_j}\|_2 \left[ \min_{j \notin S \cup S_0} \|u_{G_j}\|_2/\lambda_j \right]$$

$$\leq \sum_{j \notin S \cup S_0} \lambda_j \|u_{G_j}\|_2 \left[ \sum_{j \in S_h} \lambda_j \|u_{G_j}\|_2 / \sum_{j \in S_{k-1}} \lambda_j \|u_{G_j}\|_2 \right]$$

$$\leq \lambda_-^{-1} \sum_{k \geq 1} \sqrt{\sum_{j \in S_h} \lambda_j \|u_{G_j}\|_2} \sqrt{\sum_{j \in S_{k-1}} \lambda_j \|u_{G_j}\|_2}$$

$$\leq \lambda_-^{-1} \sum_{k \geq 1} \frac{1}{2} \left[ \sum_{j \in S_h} \lambda_j \|u_{G_j}\|_2 + \sum_{j \in S_{k-1}} \lambda_j \|u_{G_j}\|_2 \right]$$

$$\leq \lambda_-^{-1} \sum_{k \geq 0} \sum_{j \in S_h} \lambda_j \|u_{G_j}\|_2 = \lambda_-^{-1} \sum_{j \notin S} \lambda_j \|u_{G_j}\|_2.$$
Therefore
\[ n^{-1} \left| \sum_{j \notin S \cup S_0} u_G^\top X_G^\top X_G u_{G_j} \right| \leq n^{-1} \sum_{k \geq 1} \left| u_G^\top X_G^\top X_G^k u_{G^k} \right| \]
\[ \leq n^{-1} \sum_{k \geq 1} \| X_G^\top X_G^k \|_2 \| u_{G^k} \|_2 \| u_G \|_2 \]
\[ \leq \tilde{\rho}_+ \| u_G \|_2 \sum_{k \geq 1} \| u_{G^k} \|_2 \]
\[ \leq \tilde{\rho}_+ \lambda_j^{-1} \| u_G \|_2 \sum_{j \notin S} \lambda_j \| u_{G_j} \|_2. \]

Note that Lemma 2.8.4 is used to bound \( \| X_G^\top X_G \|_2 \). This proves the second inequality of the lemma.

The following lemma shows that the group \( L_1 \)-norm of the group Lasso estimator’s non-signal part is small (compared to the group \( L_1 \)-norm of the parameter estimation error in the signal part).

**Lemma 2.8.6** Let \( \text{supp}(\hat{\beta}) \in G_S \) for some \( S \subset \{1, \ldots, m\} \). Assume that for all \( j \):
\[ \lambda_j \geq 4 \rho_+ (G_j)^{1/2} \| (X_G^\top X_G)^{-1/2} X_G^\top \epsilon \|_2 / \sqrt{n}. \]

Then the solution of (2.1) satisfies:
\[ \sum_{j \notin S} \lambda_j \| \hat{\beta}_{G_j} \|_2 \leq 3 \sum_{j \in S} \lambda_j \| \hat{\beta}_{G_j} - \hat{\beta}_{G_j} \|_2. \]

**Proof** The first order condition is:
\[ 2X^\top X (\hat{\beta} - \bar{\beta}) - 2X^\top \epsilon + \sum_{j=1}^m \lambda_j n \frac{\hat{\beta}_{G_j}}{\| \hat{\beta}_{G_j} \|_2} = 0. \] (2.6)

By multiplying both sides by \( (\hat{\beta} - \bar{\beta})^\top \), we obtain
\[ 0 \geq -2(\hat{\beta} - \bar{\beta})^\top X^\top X (\hat{\beta} - \bar{\beta}) = -2(\hat{\beta} - \bar{\beta})^\top X^\top \epsilon + \sum_{j=1}^m \lambda_j n \frac{(\hat{\beta} - \bar{\beta})_{G_j}}{\| \hat{\beta}_{G_j} \|_2}. \]
Therefore
\[
\sum_{j \in S} \lambda_j \|\hat{\beta}_{G_j}\|_2 \\
\leq \sum_{j \in S} \lambda_j \|\hat{\beta}_{G_j} - \hat{\beta}_{G_j}\|_2 + 2(\hat{\beta} - \bar{\beta})^\top X^\top \epsilon/n \\
\leq \sum_{j \in S} \lambda_j \|\hat{\beta}_{G_j} - \hat{\beta}_{G_j}\|_2 + 2 \sum_{j=1}^m \rho_+(G_j)^{1/2}\|\hat{\beta} - \bar{\beta}\|_{G_j} \|X_{G_j}^\top X_{G_j}\|^{-1/2} X_{G_j}^\top \epsilon\|_2 / \sqrt{n} \\
\leq \sum_{j \in S} \lambda_j \|\hat{\beta}_{G_j} - \hat{\beta}_{G_j}\|_2 + 0.5 \sum_{j=1}^m \lambda_j \|\hat{\beta} - \bar{\beta}\|_{G_j}^2.
\]

Note that the last inequality follows from the assumption of the lemma. By simplifying the above inequality, we obtain the desired bound. This leads to the desired bound. 

The following lemma gives a parameter estimation error bound by combining the previous two lemmas.

**Lemma 2.8.7** Let supp(\(\bar{\beta}\)) \(\in G_S\) for some \(S \subset \{1, \ldots, m\}\). Consider \(\ell \geq 1\) and let \(s = |G_S| + \ell + k_0 - 1\). Define
\[
\lambda^2_j = \min \left\{ \sum_{j \in S'} \lambda^2_j : |G_S'| \geq \ell \right\},
\]
\[
\tilde{\rho}_+ = \sqrt{(\rho_+(s) - \rho_-(2s - |G_S|))(\rho_+(s - |G_S|) - \rho_-(2s - |G_S|))}.
\]

If for all \(j\):
\[
\lambda_j \geq 4\rho_+(G_j)^{1/2}\|X_{G_j}^\top X_{G_j}\|^{-1/2} X_{G_j}^\top \epsilon\|_2 / \sqrt{n},
\]
and
\[
\frac{6\tilde{\rho}_+}{\rho_-(s)} \leq \frac{\lambda_-}{\sqrt{\sum_{j \in S} \lambda^2_j}},
\]

then the solution of (2.1) satisfies:
\[
\|\hat{\beta} - \bar{\beta}\|_2 \leq \frac{1.5}{\rho_-(s)} \left( 1 + 1.5\lambda^{-1} \sqrt{\sum_{j \in S} \lambda^2_j} \right) \sqrt{\sum_{j \in S} \lambda^2_j}.
\]

**Proof** Define \(S_0\) as in Lemma 2.8.5. Let \(G = \cup_{j \in S \cup S_0} G_j\). By multiplying both sides of (2.6) by \((\hat{\beta} - \bar{\beta})_G^\top\), we obtain
\[
2(\hat{\beta} - \bar{\beta})_G^\top X_{G}^\top X(\hat{\beta} - \bar{\beta}) - 2(\hat{\beta} - \bar{\beta})_G^\top X_{G}^\top \epsilon + \sum_{j \in S \cup S_0} \lambda_j n \frac{(\hat{\beta} - \bar{\beta})_{G_j} \hat{\beta}_{G_j}}{\|\hat{\beta}_{G_j}\|_2} = 0.
\]
Similar to the proof in Lemma 2.8.6, we use the assumptions on \( \lambda_j \) to obtain:

\[
4n^{-1} (\hat{\beta} - \bar{\beta})_G^\top X_G X (\hat{\beta} - \bar{\beta}) + \sum_{j \in S_0} \lambda_j \| \hat{\beta}_{G_j} - \bar{\beta}_{G_j} \|_2^2 \leq 3 \sum_{j \in S} \lambda_j \| \hat{\beta}_{G_j} - \bar{\beta}_{G_j} \|_2^2.
\]  

(2.7)

Now, Lemma 2.8.5 implies that

\[
(\hat{\beta} - \bar{\beta})_G^\top X_G X (\hat{\beta} - \bar{\beta}) \geq (\hat{\beta} - \bar{\beta})_G^\top X_G \hat{\beta}_G - \bar{\beta}_G^\top X_G (\hat{\beta} - \bar{\beta})_G - \bar{\beta}_G + \lambda^{-1} n \| (\hat{\beta} - \bar{\beta})_G \|_2 \sum_{j \in S} \lambda_j \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2.
\]

By applying Lemma 2.8.6, we have

\[
n^{-1} (\hat{\beta} - \bar{\beta})_G^\top X_G^\top X (\hat{\beta} - \bar{\beta}) \geq \rho_-(G) \| (\hat{\beta} - \bar{\beta})_G \|_2^2 - 3 \bar{\sigma} + \lambda^{-1} n \| (\hat{\beta} - \bar{\beta})_G \|_2 \sum_{j \in S} \lambda_j \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2
\]

\[
\geq \rho_-(G) \| (\hat{\beta} - \bar{\beta})_G \|_2^2 - 3 \rho_+(G)\lambda^{-1} \sqrt{\sum_{j \in S} \lambda_j^2 \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2^2}
\]

\[
\geq 0.5 \rho_-(G) \| (\hat{\beta} - \bar{\beta})_G \|_2^2.
\]

The assumption of the lemma is used to derive the last inequality. Now plug this inequality into (2.7), we have

\[
\| (\hat{\beta} - \bar{\beta})_G \|_2^2 \leq 1.5 \rho_-(G)^{-1} \sum_{j \in S} \lambda_j \| \hat{\beta}_{G_j} - \bar{\beta}_{G_j} \|_2 \leq 1.5 \rho_-(G)^{-1} \sqrt{\sum_{j \in S} \lambda_j^2 \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2^2}.
\]

This implies

\[
\| (\hat{\beta} - \bar{\beta})_G \|_2^2 \leq 2.25 \rho_-(G)^{-2} \sum_{j \in S} \lambda_j^2.
\]

Now Lemma 2.8.5 and Lemma 2.8.6 imply that

\[
\| (\hat{\beta} - \bar{\beta})_G \|_2^2 - \| (\hat{\beta} - \bar{\beta})_G \|_2^2 \leq 0.25 \lambda^{-2} \left[ \sum_{j \in S} \lambda_j \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2^2 \right]^2
\]

\[
\leq 2.25 \lambda^{-2} \sum_{j \in S} \lambda_j \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2^2
\]

\[
\leq 2.25 \lambda^{-2} \sum_{j \in S} \lambda_j^2 \| (\hat{\beta} - \bar{\beta})_{G_j} \|_2^2.
\]

By combining the previous two displayed inequalities, we obtain the lemma.

\[\square\]

### 2.8.4 Proof of Theorem 2.5.1

Assumption 3.6.1 implies that with probability larger than \( 1 - \eta \), uniformly for all groups \( j \), we have

\[
\| (X_{G_j}^\top X_{G_j})^{-0.5} X_{G_j} (\epsilon - \mathbb{E}\epsilon) \|_2 \leq a \sqrt{k_j} + b \sqrt{\ln(m/\eta)}.
\]
It follows that with the choice of $A$, $B$, and $\lambda_j$,

$$\lambda_j \geq 4\rho_+(G_j)^{1/2}\|(X_{G_j}^\top X_{G_j})^{-1/2}X_{G_j}^\top \epsilon\|_2\sqrt{n}$$

for all $j$. Moreover, assumptions of the theorem also imply that $\tilde{\rho}_+ \leq \rho_+(s) - \rho_-(2s)$, and

$$\frac{\tilde{\rho}_+}{\rho_-(s)} \leq \frac{\rho_+(s) - \rho_-(s)}{\rho_-(s)} \leq c \leq \frac{\sqrt{\ell A^2 + \ell B^2}}{6\sqrt{2(kA^2 + gB^2)}} \leq \frac{\lambda_-}{6\sqrt{\sum_{j \in S} \lambda_j^2}}.$$ 

Note that we have used

$$\sum_{j \in S'}[A^2 k_j + B^2] \leq n \sum_{j \in S'} \lambda_j^2 \leq 2 \sum_{j \in S'}[A^2 k_j + B^2].$$

Therefore the conditions of Lemma 2.8.7 are satisfied. Its conclusion implies that

$$\|\hat{\beta} - \tilde{\beta}\|_2 \leq \frac{1.5}{\rho_-(s)} \left(1 + \frac{1.5}{4c} \right) \sqrt{\sum_{j \in S} \lambda_j^2} \|\tilde{\epsilon}\|_2 \leq \frac{1.5}{\rho_-(s)} \left(1 + \frac{1}{4c} \right) \sqrt{2(A^2 k + B^2 g)/n}. $$

This proves the theorem.
Chapter 3

Learning With Structured Sparsity

This chapter generalizes the group sparsity idea to structured sparsity by allowing arbitrary structures on the feature set. A general theory is developed for learning with structured sparsity, based on the notion of coding complexity associated with the structure. It is shown that if the coding complexity of the target signal is small, then one can achieve improved performance by using coding complexity regularization methods, which generalizes the standard sparse regularization. This work was presented under a slightly modified form in [66].

3.1 Introduction

The related work in the previous chapters show that the structured sparsity enables better sparse learning from fewer samples. However, none of existing algorithms for structured sparsity do have: 1) explicit bounds on the minimal number of measurements; 2) provable recovery performance guarantees from noise measurements; and 3) a general scheme to flexibly describe different structures.

In this chapter, a new framework will be proposed to satisfy all these conditions based on the new theorem, which is a natural extension of the standard sparsity in statistical learning and compressive sensing. By allowing arbitrary structures on the feature set, this new framework generalizes the group sparsity idea that has become popular in recent years. A general theory is developed for learning with structured sparsity, based on the notion of coding complexity associated with the structure. It is shown that if the coding complexity of the target signal is small, then one can achieve improved performance by using coding complexity regularization methods, which generalize the standard sparse regularization. Moreover, a structured greedy algorithm is proposed to efficiently solve the structured sparsity problem. It is shown that the greedy algorithm approximately solves the coding complexity optimization problem under appropriate conditions. If meaningful structures exist, we show that our new framework can take advantage of such structures to improve the standard sparse learning.
3.2 Related Work

The idea of using group structure to achieve better results has received much attention. For example, group sparsity has been considered for simultaneous sparse approximation [140] and multi-task compressive sensing [72] from the Bayesian hierarchical modeling point of view. Under the Bayesian hierarchical model framework, data from all sources contribute to the estimation of hyper-parameters in the sparse prior model. The shared prior can then be inferred from multiple sources. He et al. recently extend the idea to the tree sparsity in the Bayesian framework [55, 54]. Although the idea can be justified using standard Bayesian intuition, there are no theoretical results showing how much better (and under what kind of conditions) the resulting algorithms perform.

In the statistical literature, the Lasso has been extended to the group Lasso while there exist group/block structured dependences among the sparse coefficients [148]. The sparse learning problem is then solved with the $L_{2,1}$ norm regularization. Recently, a more efficient spectral projected-gradient algorithm [14] is proposed to solve the group Lasso problem, which is based on a linear-time algorithm for Euclidean projection. The algorithm is more suitable for large scale group sparse learning. Elitist Lasso is proposed to the same problem with the $L_{1,2}$ norm regularization in [78]. However, these works are short of theoretical analysis for both bounds on the minimal number of measurements and the recovery guarantees. Although some theoretical results were developed in [96], neither showed that group Lasso is superior to the standard Lasso. The authors of [76] show that group Lasso can be superior to standard Lasso when each group is an infinite dimensional kernel, by relying on the fact that meaningful analysis can be obtained for kernel methods in infinite dimension. In [101], the authors consider a special case of group Lasso in the multi-task learning scenario, and show that the number of samples required for recovering the exact support set is smaller for group Lasso under appropriate conditions. In [65], we develop a theory for group Lasso using a concept called strong group sparsity and show that group Lasso is superior to standard Lasso for strongly group-sparse signals, which provides a convincing theoretical justification for using group structured sparsity. However, these algorithms can only handle the cases where the group setting are static and fixed once for all. Different regularization terms are proposed to extend the group Lasso, such as elastic net penalty [158], composite absolute penalty (CAP) [155] and mixed norm penalty [148, 78, 77, 79, 84]. The hierarchy structure can be well exploited by a mixed norm penalty [77, 79, 84]. One of their shortcomings is the independence of the static groups. It means that a given coefficient can not belong to different groups, which make them too rigid for some
practical data. In [155], composite absolute penalty is proposed to handle both cases of none-overlapping groups and overlapping groups. However, no theoretical proof is given to the CAP. In [80], Kowalski et al. generalized the mixed norm penalty to structured shrinkage, which can identify the structured significance maps and thus can handle the case of the overlapping groups. However, the structured Shrinkage operations have not been theoretically proved to convergence to a fixed point.

Tropp et al. proposes the Simultaneous OMP for group sparsity problem in [128]. In [126], the convex relaxation solution is proposed for the same problem. The theoretical analysis is also provided. However, they assume that the group structures are known before recovery. Moreover, they only consider the case where all groups share a common nonzero coefficient support set. In [121], the authors attempt to derive a bound on the number of samples needed to recover block sparse signals, where the sparse coefficients in each block are either all zero or all nonzero. This corresponds to the case of group sparsity with equal size groups. It is not clear how to extend their work to the general structured sparsity. Daudet [30] combines the sparse representation in overcomplete spaces and structured representation in image coding for decompositions of audio signals. The basis of the overcomplete spaces are the unions of a basis of Modified Discrete Cosine Transform (MDCT) and a basis of Dyadic Wavelet Transform (DWT). The priors of the tonal structure in MDCT and the transient structure in DWT are used for the sparse decomposition. However, no theorems and proofs are provided for the proposed algorithm. Thus, it is not clear how to generalize the propose algorithm for more general cases. To obtain the global optimization results, Grimm et al. [50] investigate positive polynomials with structured sparsity in which every monomial involves only variables from blocks of variables satisfying certain overlap conditions. They derive the proof for Lasserre’s theorem on the existence of sums of squares certificates with respect to the block structure. In recent work [7], the restricted amplification property is proposed for the model based sparse data instead of restricted isometry property (RIP) for the traditional sparse data. It was shown to provide a sufficient condition for robust recovery of model-compressible data whose structured sparsity is constrained by the given model. It is furthermore shown that simple modifications of the CoSaMP algorithm [97] and of iterative hard thresholding [18] yield recovery algorithms for the model-based sparse data. Their model based framework offers a systematic method for designing provably efficient CS recovery algorithms for data belonging to these models. The tree sparsity model and block sparsity model have been examined as two instances of their framework. In [19], this model based framework has been successfully applied to the data model motivated by neuronal spike trains. However, no general scheme was proposed to describe the
model in their framework. Thus, different schemes have to be designed to describe different data model before utilizing their model based framework. Therefore, it remains as a problem how to adaptively design a general scheme to efficiently and effectively describe the model.

The above introduction shows that the structured sparsity enables better sparse learning from fewer samples. However, none of existing algorithms for structured sparsity do have: 1) explicit bounds on the minimal number of measurements; 2) provable recovery performance guarantees from noise measurements; and 3) a general scheme to flexibly describe different structures. In this chapter, a new framework will be proposed to satisfy all these conditions based on the new theorem.

3.3 Structured Sparsity

In structured sparsity, not all sparse patterns are equally alike. For example, in group sparsity, coefficients within the same group are more likely to be zeros or nonzeros simultaneously. This means that if a sparse coefficient’s support set is consistent with the underlying group structure, then it is more likely to occur, and hence incurs a smaller penalty in learning. One contribution of this work is to formulate how to define structure on top of sparsity, and how to penalize each sparsity pattern.

In order to formalize the idea, we denote by \( \mathcal{I} = \{1, \ldots, p\} \) the index set of the coefficients. We assign a cost \( \text{cl}(F) \) to any sparse subset \( F \subset \{1, \ldots, p\} \). In structured sparsity, \( \text{cl}(F) \) is an upper bound of the coding length of \( F \) (number of bits needed to represent \( F \) by a computer program) in a pre-chosen prefix coding scheme. It is a well-known fact in information theory that mathematically, the existence of such a coding scheme is equivalent to \( \sum_{F \subset \mathcal{I}, F \neq \emptyset} 2^{-\text{cl}(F)} \leq 1 \).

From the Bayesian statistics point of view, \( 2^{-\text{cl}(F)} \) can be regarded as a lower bound of the probability of \( F \). The probability model of structured sparse learning is thus: first generate the sparsity pattern \( F \) according to probability \( 2^{-\text{cl}(F)} \); then generate the coefficients in \( F \).

**Definition 3.3.1** A cost function \( \text{cl}(F) \) defined on subsets of \( \mathcal{I} \) is called a coding length (in base-2) if

\[
\sum_{F \subset \mathcal{I}, F \neq \emptyset} 2^{-\text{cl}(F)} \leq 1.
\]

We give \( \emptyset \) a coding length 0. The corresponding structured sparse coding complexity of \( F \) is defined as

\[
c(F) = |F| + \text{cl}(F).
\]
A coding length $\text{cl}(F)$ is sub-additive if $\text{cl}(F \cup F') \leq \text{cl}(F) + \text{cl}(F')$, and a coding complexity $c(F)$ is sub-additive if $c(F \cup F') \leq c(F) + c(F')$.

Clearly if $\text{cl}(F)$ is sub-additive, then the corresponding coding complexity $c(F)$ is also sub-additive. Based on the structured coding complexity of subsets of $\mathcal{I}$, we can now define the structured coding complexity of a sparse coefficient vector $\hat{\beta} \in \mathbb{R}^p$.

**Definition 3.3.2** Giving a coding complexity $c(F)$, the structured sparse coding complexity of a coefficient vector $\hat{\beta} \in \mathbb{R}^p$ is

$$c(\hat{\beta}) = \min \{c(F) : \text{supp}(\hat{\beta}) \subset F\}.$$

Later, we will show that if a coefficient vector $\hat{\beta}$ has a small coding complexity $c(\hat{\beta})$, then $\hat{\beta}$ can be effectively learned, with good in-sample prediction performance (in statistical learning) and reconstruction performance (in compressive sensing). In order to see why the definition requires adding $|F|$ to $\text{cl}(F)$, we consider the generative model for structured sparsity mentioned earlier. In this model, the number of bits to encode a sparse coefficient vector is the sum of the number of bits to encode $F$ (which is $\text{cl}(F)$) and the number of bits to encode nonzero coefficients in $F$ (this requires $O(|F|)$ bits up to a fixed precision). Therefore the total number of bits required is $\text{cl}(F) + O(|F|)$. This result can be translated into a statistical estimation result: without additional regularization, the learning complexity for least squares regression within any fixed support set $F$ is $O(|F|)$. By adding the model selection complexity $\text{cl}(F)$ for each support set $F$, we obtain an overall statistical estimation complexity of $O(\text{cl}(F) + |F|)$. While the idea of using coding based penalization resembles minimum description length (MDL), the actual penalty we obtain for structured sparsity problems is different from the standard MDL penalty for model selection. This difference is important, and thus in order to prevent confusion, we avoid using MDL in our terminology.

### 3.4 General Coding Scheme

We introduce a general coding scheme called **block coding**. The basic idea of block coding is to define a coding scheme on a small number of base blocks (a block is a subset of $\mathcal{I}$), and then define a coding scheme on all subsets of $\mathcal{I}$ using these base blocks.

Consider a subset $\mathcal{B} \subseteq 2^\mathcal{I}$. That is, each element (a block) of $\mathcal{B}$ is a subset of $\mathcal{I}$. We call $\mathcal{B}$ a block set if $\mathcal{I} = \cup_{B \in \mathcal{B}} B$ and all single element sets $\{j\}$ belong to $\mathcal{B}$ ($j \in \mathcal{I}$). Note that $\mathcal{B}$ may contain additional non single-element blocks. The requirement of $\mathcal{B}$ containing all single
element sets is for convenience, as it implies that every subset $F \subset \mathcal{I}$ can be expressed as the union of blocks in $\mathcal{B}$.

Let $c_0$ be a code length on $\mathcal{B}$:
$$
\sum_{B \in \mathcal{B}} 2^{-c_0(B)} \leq 1,
$$
we define $c(B) = c_0(B) + 1$ for $B \in \mathcal{B}$. It not difficult to show that the following cost function on $F \subset \mathcal{I}$ is a code-length
$$
cl(F) = \min \left\{ \sum_{j=1}^{b} c(B_j) : F = \bigcup_{j=1}^{b} B_j \ (B_j \in \mathcal{B}) \right\}.
$$
This is a coding length because
$$
\sum_{F \subset \mathcal{I}, F \neq \emptyset} 2^{-c(F)} \leq \sum_{b \geq 1} \sum_{\{B_i\} \in \mathcal{B}^b} 2^{-\sum_{i=1}^{b} c(B_i)} \\
\leq \sum_{b \geq 1} \prod_{\ell=1}^{b} \sum_{B \in \mathcal{B}} 2^{-c(B)} \leq \sum_{b \geq 1} 2^{-b} = 1.
$$
It is obvious that block coding is sub-additive.

The main purpose of introducing block coding is to design computational efficient algorithms based on the block structure. In particular, we consider a structured greedy algorithm that can take advantage of block structures. In the structured greedy algorithm, instead of searching over all subsets of $\mathcal{I}$ up to a fixed coding complexity $s$ (exponential in $s$ number of such subsets), we greedily add blocks from $\mathcal{B}$ one at a time. Each search problem over $\mathcal{B}$ can be efficiently performed because $\mathcal{B}$ is supposed to contain only a computationally manageable number of base blocks. Therefore the algorithm is computationally efficient. Concrete structured sparse coding examples described below can be efficiently approximated by block coding. This means the greedy algorithm can efficiently handle such structure.

**Standard sparsity**

A simple coding scheme is to code each subset $F \subset \mathcal{I}$ of cardinality $k$ using $k \log_2(2p)$ bits, which corresponds to block coding with $\mathcal{B}$ consisted only of single element sets, and each base block has a coding length $\log_2 p$. This corresponds to the complexity for the standard sparse learning.

**Group sparsity**

The concept of group sparsity has been introduced in various recent work, such as the group Lasso in [148]. Consider a partition of $\mathcal{I} = \cup_{j=1}^{m} G_j$ to $m$ disjoint groups. Let $\mathcal{B}_G$ contain the $m$
Figure 3.1: Examples of structured sparsity, where black nodes are selected variables in all available nodes: (a) standard sparsity; (b) group sparsity; (c) line-graph sparsity; (d) tree sparsity and wavelet tree sparsity; (e) grid sparsity and graph sparsity in background subtracted images.
groups $G_j$, and $\mathcal{B}_1$ contain $p$ single element blocks. The strong group sparsity coding scheme is to give each element in $\mathcal{B}_1$ a code-length $c_{l_0}$ of $\infty$, and each element in $\mathcal{B}_G$ a code-length $c_{l_0}$ of $\log_2 m$. Then the block coding scheme with blocks $\mathcal{B} = \mathcal{B}_G \cup \mathcal{B}_1$ leads to group sparsity, which only looks for signals consisted of the groups. The resulting coding length is: $c_l(B) = g \log_2(2m)$ if $B$ can be represented as the union of $g$ disjoint groups $G_j$; and $c_l(B) = \infty$ otherwise. Note that if the signal can be expressed as the union of $g$ groups, and each group size is $k_0$, then the group coding length $g \log_2(2m)$ can be significantly smaller than the standard sparsity coding length of $g k_0 \log_2(p)$. As we shall see later, the smaller coding complexity implies better learning behavior, which is essentially the advantage of using group sparse structure.

Hierarchical sparsity

One may also create a hierarchical group structure. A simple example is wavelet coefficients of a signal [89]. Another simple example is a binary tree with the variables as leaves, which we describe below. Each internal node in the tree is associated with three options: left child only, right child only, and both children; each option can be encoded in $\log_2 3$ bits.

Given a subset $F \subset \mathcal{I}$, we can go down from the root of the tree, and at each node, decide whether only left child contains elements of $F$, or only right child contains elements of $F$, or both children contain elements of $F$. Therefore the coding length of $F$ is $\log_2 3$ times the total number of internal nodes leading to elements of $F$. Since each leaf corresponds to no more than $\log_2 p$ internal nodes, the total coding length is no worse than $\log_2 3 \log_2 p |F|$. However, the coding length can be significantly smaller if nodes are close to each other or are clustered.

In the extreme case, when the nodes are consecutive, we have $O(|F| + \log_2 p)$ coding length. More generally, if we can order elements in $F$ as $F = \{j_1, \ldots, j_q\}$, then the coding length can be bounded as $c_l(F) = O(|F| + \log_2 p + \sum_{s=2}^{q} \log_2 \min_{\ell<s} |j_s - j_\ell|)$.

If all internal nodes of the tree are also variables in $\mathcal{I}$ (for example, in the case of wavelet decomposition), then one may consider feature set $F$ with the following property: if a node is selected, then its parent is also selected. This requirement is very effective in wavelet compression, and often referred to as the zero-tree structure [113]. Similar requirements have also been applied in statistics [155] for variable selection. The argument presented in this section shows that if we require $F$ to satisfy the zero-tree structure, then its coding length is at most $O(|F|)$, without any explicit dependency on the dimensionality $p$. This is because one does not have to reach a leaf node.

The tree-based coding scheme discussed in this section can be approximated by block coding.
using no more than $p^{1+\delta}$ base blocks ($\delta > 0$). The idea is similar to that of the image coding example in the more general graph sparsity scheme which we discuss next.

The wavelet coefficients of a signal are well known to have a tree-graph structure, which has been widely used for compressing natural images and is a special case of graph sparsity. Each wavelet coefficient of the signal is connected to its parent coefficient and its child coefficients. The wavelet coefficients of 1D signals have a binary tree connected-graph structure while the wavelet coefficients of 2D images have a quad-tree connected-graph structure.

**Graph sparsity**

We generalize the group sparsity idea that employs a directed graph structure $G$ on $\mathcal{I}$. Each element of $\mathcal{I}$ is a node of $G$ but $G$ may contain additional nodes. For simplicity, we assume $G$ contains a starting node not in $\mathcal{I}$. At each node $v \in G$, we define coding length $c_v(S)$ on the neighborhood $N_v$ of $v$ (that contains the empty set), as well as any other single node $u \in G$ with $c_v(u)$, such that $\sum_{S \subset N_v} 2^{-c_v(S)} + \sum_{u \in G} 2^{-c_v(u)} \leq 1$. To encode $F \subset G$, we start with the active set containing only the starting node, and finish when the set becomes empty. At each node $v$ before termination, we may either pick a subset $S \subset N_v$, with coding length $c_v(S)$, or a node in $u \in G$, with coding length $c_v(u)$, and then put the selection into the active set. We then remove $v$ from the active set (once $v$ is removed, it does not return to the active set anymore). This process is continued until the active set becomes empty.

As a concrete example, we consider image processing problem, where each image is a rectangle of pixels (nodes); each pixel is connected to four adjacent pixels, which forms the underlying graph structure. At each pixel, the number of subsets in its neighborhood is $2^4 = 16$ (including the empty set), with a coding length $c_v(S) = 5$ each; we also encode all other pixels in the image with random jumping, each with a coding length $1 + \log_2 p$. Using this scheme, we can encode each connected region $F$ by no more than $\log_2 p + 5|F|$ bits by growing the region from a single point in the region. Therefore if $F$ is composed of $g$ connected regions, then the coding length is $g \log_2 p + 5|F|$, which can be significantly better than standard sparse coding length of $|F| \log_2 p$. This example shows that the general graph coding scheme presented here favors connected regions (that is, nodes that are grouped together with respect to the graph structure). This scheme can be efficiently approximated with block coding as follows: we consider relatively small sized base blocks consisted of nodes that are close together with respect to the graph structure, and then use the induced block coding scheme to approximate the graph coding.
3.5 Algorithms for Structured Sparsity

The following algorithm is a natural extension of $L_0$ regularization to structured sparsity problems. It penalizes the coding complexity instead of the cardinality (sparsity) of the feature set.

$$\hat{\beta}_{\text{constr}} = \arg\min_{\beta \in \mathbb{R}^p} \hat{Q}(\beta) \text{ subject to } c(\beta) \leq s. \quad (3.1)$$

The optimization of (3.1) is generally hard. There are two common approaches to alleviate this problem. One is convex relaxation ($L_1$ regularization to replace $L_0$ regularization for standard sparsity); the other is forward greedy algorithm. We do not know any extensions of $L_1$ regularization like convex relaxation that can handle general structured sparsity formulations. However, one can extend greedy algorithm by using a block structure. We call the resulting procedure structured greedy algorithm (see Algorithm 1), which approximately solves (3.1).

Algorithm 1 Structured Greedy Algorithm (StructOMP)

1: **Input:** $(X, y), \mathcal{B} \subset 2^I, s > 0$
2: **Output:** $F^{(k)}$ and $\beta^{(k)}$
3: let $F^{(0)} = \emptyset$ and $\beta^{(0)} = 0$
4: for all $K = 1, \ldots$ do
5:     select $B^{(k)} \in \mathcal{B}$ to maximize progress (*)
6:     let $F^{(k)} = F^{(k-1)} \cup B^{(k)}$
7:     let $\beta^{(k)} = \arg\min_{\beta \in \mathbb{R}^p} \hat{Q}(\beta)$
        subject to $\text{supp}(\beta) \subset F^{(k)}$
8:     if $(c(\beta^{(k)}) > s)$ break
9: end for

In Algorithm 1, we are given a set of blocks $\mathcal{B}$ that contains subsets of $I$. Instead of searching all subsets $F \subset I$ up to a certain complexity $|F| + c(F)$, which is computationally infeasible, we search only the blocks restricted to $\mathcal{B}$. It is assumed that searching over $\mathcal{B}$ is computationally manageable. At each step (*), we try to find a block from $\mathcal{B}$ to maximize progress. It is thus necessary to define a quantity that measures progress. Our idea is to approximately maximize the gain ratio:

$$\lambda^{(k)} = \frac{\hat{Q}(\beta^{(k-1)}) - \hat{Q}(\beta^{(k)})}{c(\beta^{(k)}) - c(\beta^{(k-1)})},$$

which measures the reduction of objective function per unit increase of coding complexity. This greedy criterion is a natural generalization of the standard greedy algorithm, and is essential in our analysis. For least squares regression, we can approximate $\lambda^{(k)}$ using the following definition

$$\phi(B) = \frac{\|P_{B - F^{(k-1)}}(X\beta^{(k-1)} - y)\|_2^2}{c(B \cup F^{(k-1)}) - c(F^{(k-1)})}, \quad (3.2)$$

where $P_{\mathcal{F}} = X_{\mathcal{F}}(X_{\mathcal{F}}^\top X_{\mathcal{F}})^{-1}X_{\mathcal{F}}^\top$ is the projection matrix to the subspaces generated by columns.
of \( X_F \). We then select \( B^{(k)} \) so that

\[
\phi(B^{(k)}) \geq \gamma \max_{B \in \mathcal{B}} \phi(B),
\]

where \( \gamma \in (0, 1] \) is a fixed approximation ratio that specifies the quality of approximate optimization.

### 3.6 Theory of Structured Sparsity

#### 3.6.1 Assumptions

We assume sub-Gaussian noise as follows.

**Assumption 3.6.1** Assume that \( \{y_i\}_{i=1,...,n} \) are independent (but not necessarily identically distributed) sub-Gaussians: there exists a constant \( \sigma \geq 0 \) such that \( \forall i \) and \( \forall t \in \mathbb{R} \),

\[
\mathbb{E}_{y_i} e^{t(y_i - \mathbb{E}y_i)} \leq e^{\sigma^2 t^2/2}.
\]

We also need to generalize sparse eigenvalue condition, used in the modern sparsity analysis. It is related to (and weaker than) the RIP (restricted isometry property) assumption [22] in the compressive sensing literature. This definition takes advantage of coding complexity, and can be also considered as (a weaker version of) structured RIP. We introduce a definition.

**Definition 3.6.1** For all \( F \subset \{1, \ldots, p\} \), define

\[
\rho_-(F) = \inf \left\{ \frac{1}{n} \|X\beta\|_2^2/\|\beta\|_2^2 : \text{supp}(\beta) \subset F \right\},
\]

\[
\rho_+(F) = \sup \left\{ \frac{1}{n} \|X\beta\|_2^2/\|\beta\|_2^2 : \text{supp}(\beta) \subset F \right\}.
\]

Moreover, for all \( s > 0 \), define

\[
\rho_-(s) = \inf \{\rho_-(F) : F \subset \mathcal{I}, c(F) \leq s\},
\]

\[
\rho_+(s) = \sup \{\rho_+(F) : F \subset \mathcal{I}, c(F) \leq s\}.
\]

In the theoretical analysis, we need to assume that \( \rho_-(s) \) is not too small for some \( s \) that is larger than the signal complexity. Since we only consider eigenvalues for submatrices with small cost \( c(\bar{\beta}) \), the sparse eigenvalue \( \rho_-(s) \) can be significantly larger than the corresponding ratio for standard sparsity (which will consider all subsets of \( \{1, \ldots, p\} \) up to size \( s \)). For example, for random projections used in compressive sensing applications, the coding length \( c(\text{supp}(\bar{\beta})) \) is \( O(k \ln p) \) in standard sparsity, but can be as low as \( c(\text{supp}(\bar{\beta})) = O(k) \) in structured sparsity (if we can guess \( \text{supp}(\bar{\beta}) \) approximately correctly. Therefore instead of requiring \( n = O(k \ln p) \)
samples, we requires only $O(k + \text{cl}(\text{supp}(\hat{\beta})))$. The difference can be significant when $p$ is large and the coding length $\text{cl}(\text{supp}(\hat{\beta})) \ll k \ln p$.

The theorem implies that the structured RIP condition is satisfied with sample size $n = O((k/k_0) \ln(p/k_0))$ in group sparsity rather than $n = O(k \ln(p))$ in standard sparsity.

**Theorem 3.6.1 (Structured-RIP)** Suppose that elements in $X$ are iid standard Gaussian random variables $N(0, 1)$. For any $t > 0$ and $\delta \in (0, 1)$, let

$$n \geq \frac{8}{\delta^2} \ln 3 + t + s \ln(1 + 8/\delta).$$

Then with probability at least $1 - e^{-t}$, the random matrix $X \in \mathbb{R}^{n \times p}$ satisfies the following structured-RIP inequality for all vector $\bar{\beta} \in \mathbb{R}^p$ with coding complexity no more than $s$:

$$\left(1 - \delta\right)\|\bar{\beta}\|_2 \leq \frac{1}{\sqrt{n}}\|X\bar{\beta}\|_2 \leq (1 + \delta)\|\bar{\beta}\|_2.$$  \tag{3.3}

### 3.6.2 Coding complexity regularization

**Theorem 3.6.2** Suppose that Assumption 3.6.1 is valid. Consider any fixed target $\bar{\beta} \in \mathbb{R}^p$. Then with probability exceeding $1 - \eta$, for all $\lambda \geq 0, \epsilon \geq 0, \hat{\beta} \in \mathbb{R}^p$ such that:

$$Q(\hat{\beta}) \leq Q(\bar{\beta}) + \epsilon,$$

we have

$$\|X\hat{\beta} - \mathbb{E}y\|_2 \leq \|X\bar{\beta} - \mathbb{E}y\|_2 + \sigma \sqrt{2 \ln(6/\eta)} + 2\Gamma,$$

$$\Gamma = (7.4\sigma^2 c(\hat{\beta}) + 2.4\sigma^2 \ln(6/\eta) + \epsilon)^{1/2}.$$

Moreover, if the coding scheme $c(\cdot)$ is sub-additive, then

$$n\rho - (c(\bar{\beta}) + c(\hat{\beta}))\|\hat{\beta} - \bar{\beta}\|_2^2 \leq 10\|X\bar{\beta} - \mathbb{E}y\|_2^2 + \Delta,$$

$$\Delta = 37\sigma^2 c(\hat{\beta}) + 29\sigma^2 \ln(6/\eta) + 2.5\epsilon.$$

This theorem immediately implies the following result for (3.1): $\forall \hat{\beta}$ such that $c(\hat{\beta}) \leq s$,

$$\frac{1}{\sqrt{n}}\|X\hat{\beta}_\text{constr} - \mathbb{E}y\|_2 \leq \frac{1}{\sqrt{n}}\|X\bar{\beta} - \mathbb{E}y\|_2 + \Lambda,$$

$$\Lambda = \frac{\sigma}{\sqrt{n}} \sqrt{2 \ln(6/\eta)} + \frac{2\sigma}{\sqrt{n}} (7.4s + 4.7 \ln(6/\eta))^{1/2},$$

$$\|\hat{\beta}_\text{constr} - \bar{\beta}\|_2^2 \leq \frac{1}{\rho - (s + c(\bar{\beta}))n} \left[10\|X\bar{\beta} - \mathbb{E}y\|_2^2 + \Pi\right],$$

$$\Pi = 37\sigma^2 s + 29\sigma^2 \ln(6/\eta).$$

In compressive sensing applications, we take $\sigma = 0$, and we are interested in recovering $\hat{\beta}$ from random projections. For simplicity, we let $X\hat{\beta} = \mathbb{E}y = y$, and our result shows that the
constrained coding complexity penalization method achieves exact reconstruction \( \hat{\beta}_{\text{constr}} = \beta \) as long as \( \rho_-(2c(\hat{\beta})) > 0 \) (by setting \( s = c(\hat{\beta}) \)). According to Theorem 3.6.1, this is possible when the number of random projections (sample size) reaches \( n = O(2c(\hat{\beta})) \). This is a generalization of corresponding results in compressive sensing [22]. As we have pointed out earlier, this number can be significantly smaller than the standard sparsity requirement of \( n = O(\|\bar{\beta}\|_0 \ln p) \), when the structure imposed is meaningful.

### 3.6.3 Structured greedy algorithm

**Definition 3.6.2** Given \( \mathcal{B} \subset 2^I \), define

\[
\rho_0(\mathcal{B}) = \max_{B \in \mathcal{B}} \rho_+(B), \quad c_0(\mathcal{B}) = \max_{B \in \mathcal{B}} c(B)
\]

and

\[
c(\bar{\beta}, \mathcal{B}) = \min \sum_{j=1}^b c(\bar{B}_j), \quad \text{supp}(\bar{\beta}) \subset \bigcup_{j=1}^b \bar{B}_j \ (\bar{B}_j \in \mathcal{B}).
\]

The following theorem shows that if \( c(\bar{\beta}, \mathcal{B}) \) is small, then one can use the structured greedy algorithm to find a coefficient vector \( \beta^{(k)} \) that is competitive to \( \bar{\beta} \), and the coding complexity \( c(\beta^{(k)}) \) is not much worse than that of \( c(\bar{\beta}, \mathcal{B}) \). This implies that if the original coding complexity \( c(\bar{\beta}) \) can be approximated by block complexity \( c(\bar{\beta}, \mathcal{B}) \), then we can approximately solve (3.1).

**Theorem 3.6.3** Suppose the coding scheme is sub-additive. Consider \( \bar{\beta} \) and \( \epsilon \) such that \( \epsilon \in (0, \|y\|_2^2 - \|X\bar{\beta} - y\|_2^2] \) and

\[
s \geq \frac{\rho_0(\mathcal{B}) c(\bar{\beta}, \mathcal{B})}{\gamma \rho_-(s + c(\bar{\beta}))} \ln \frac{\|y\|_2^2 - \|X\bar{\beta} - y\|_2^2}{\epsilon}.
\]

Then at the stopping time \( k \), we have

\[
\hat{Q}(\beta^{(k)}) \leq \hat{Q}(\bar{\beta}) + \epsilon.
\]

By Theorem 3.6.2, the result in Theorem 3.6.3 implies that

\[
\|X\beta^{(k)} - Ey\|_2 \leq \|X\bar{\beta} - Ey\|_2 + \sigma \sqrt{2\ln(6/\eta)} + \Lambda,
\]

\[
\Lambda = 2\sigma(7.4(s + c_0(\mathcal{B}))) + 4.7\ln(6/\eta) + \epsilon/\sigma^2)^{1/2},
\]

\[
\|\beta^{(k)} - \bar{\beta}\|_2^2 \leq \frac{[10\|X\bar{\beta} - Ey\|_2^2 + \Pi]}{\rho_-(s + c_0(\mathcal{B}) + c(\bar{\beta}))n},
\]

\[
\Pi = 37\sigma^2(s + c_0(\mathcal{B})) + 29\sigma^2 \ln(6/\eta) + 2.5\epsilon.
\]

The result shows that in order to approximate a signal \( \bar{\beta} \) up to \( \epsilon \), one needs to use coding complexity \( O(\ln(1/\epsilon))c(\bar{\beta}, \mathcal{B}) \). If \( \mathcal{B} \) contains small blocks and their sub-blocks with equal coding
length, and the coding scheme is block coding generated by $B$, then $c(\hat{\beta}, B) = c(\hat{\beta})$. In this case we need $O(s \ln(1/\epsilon))$ to approximate a signal with coding complexity $s$.

In order to get rid of the $O(\ln(1/\epsilon))$ factor, backward greedy strategies can be employed, as shown in various recent work such as [152]. For simplicity, we will not analyze such strategies here. However, in the following, we present an additional convergence result for structured greedy algorithm that can be applied to weakly sparse $p$-compressible signals common in practice. It is shown that the $\ln(1/\epsilon)$ can be removed for such weakly sparse signals. More precisely, we introduce the following concept of weakly sparse compressible target that generalizes the corresponding concept of compressible signal in standard sparsity from the compressive sensing literature [32].

Definition 3.6.3 The target $Ey$ is $(a, q)$-compressible with respect to block $B$ if there exist constants $a, q > 0$ such that for each $s > 0$, $\exists \bar{\beta}(s)$ such that $c(\bar{\beta}(s), B) \leq s$ and

$$\frac{1}{n} \|Xz(\bar{\beta}(s)) - Ey\|^2_2 \leq as^{-q}.$$  

Theorem 3.6.4 Suppose that the target is $(a, q)$-compressible with respect to $B$. Then with probability $1 - \eta$, at the stopping time $k$, we have

$$\hat{Q}(\beta(k)) \leq \hat{Q}(\bar{\beta}(s')) + 2na/s'^q + 2\sigma^2[\ln(2/\eta) + 1],$$

where

$$s' \leq \frac{s \gamma}{(10 + 3q)\rho_0(B)} \min_{u \leq s'} \rho_-(s + c(\bar{\beta}(u))).$$

This result shows that we can approximate a compressible signal of complexity $s'$ with complexity $s = O(qs')$ using greedy algorithm. This means the greedy algorithm obtains optimal rate for weakly-sparse compressible signals. The sample complexity suffers only a constant factor $O(q)$. Combine this result with Theorem 3.6.2, and take union bound, we have with probability $1 - 2\eta$, at stopping time $k$:

$$\frac{1}{\sqrt{n}} \|Xz(\bar{\beta}(k)) - Ey\|_2 \leq \sqrt{\frac{a}{s'^q} + \sigma^2 \frac{2\ln(6/\eta)}{n}} + 2\sigma \sqrt{\Lambda},$$

$$\Lambda = \frac{7.4(s + c_0(B)) + 6.7 \ln(6/\eta)}{n} + \frac{2a}{n^2 s'^q},$$

$$\|\beta(k) - \bar{\beta}\|^2_2 \leq \frac{1}{\rho_-(s + s' + c_0(B))} \left[ \frac{15a}{s'^q} + \frac{\Pi}{n} \right],$$

$$\Pi = 37\sigma^2(s + c_0(B)) + 34\sigma^2 \ln(6/\eta).$$

Given a fixed $n$, we can obtain a convergence result by choosing $s$ (and thus $s'$) to optimize the right hand side. The resulting rate is optimal for the special case of standard sparsity,
which implies that the bound has the optimal form for structured $q$-compressible targets. In particular, in compressive sensing applications where $\sigma = 0$, we obtain when samples size reaches $n = O(qs')$, the reconstruction performance is

$$
\|\hat{\beta}^{(k)} - \beta\|^2 = O(a/s'^q),
$$

which matches that of the constrained coding complexity regularization method in (3.1) up to a constant $O(q)$.

### 3.7 Experiments

All experiments are conducted on a 2.4GHz PC in Matlab environment.

#### 3.7.1 Simulated 1D Signals with Line-Structured Sparsity

![Figure 3.2: Recovery results of 1D signal with strongly line-structured sparsity. (a) original data; (b) recovered results with OMP (error is 0.9921); (c) recovered results with Lasso (error is 0.8660); (d) recovered results with Group Lasso (error is 0.4832 with group size $gs=2$); (e) recovered results with Group Lasso (error is 0.4832 with group size $gs=4$); (f) recovered results with Group Lasso (error is 0.2646 with group size $gs=8$); (g) recovered results with Group Lasso (error is 0.3980 with group size $gs=16$); (h) recovered results with StructOMP (error is 0.0246).](image)

In the first experiment, we randomly generate a 1D structured sparse signal with values $\pm 1$, where data dimension $p = 512$, sparsity number $k = 64$ and group number $g = 4$. The support set of these signals is composed of $g$ connected regions. Here, each component of the sparse coefficient is connected to two of its adjacent components, which forms the underlying
Figure 3.3: Recovery performance: (a) Recovery Error vs. Sample Size Ratio ($n/k$); (b) CPU Time vs. Sample Size Ratio ($n/k$).
graph structure. The graph sparsity concept introduced earlier is used to compute the coding length of sparsity patterns in StructOMP. The projection matrix $X$ is generated by creating an $n \times p$ matrix with i.i.d. drawn from a standard Gaussian distribution $N(0, 1)$. For simplicity, the rows of $X$ are normalized to unit magnitude. Zero-mean Gaussian noise with standard deviation $\sigma = 0.01$ is added to the measurements. Our task is to compare the recovery performance of StructOMP to those of OMP, Lasso and group Lasso for these structured sparsity signals under the framework of compressive sensing.

Figure 4.3 shows one instance of generated signal and the corresponding recovered results by different algorithms when $n = 160$. Since the sample size $n$ is not big enough, OMP and Lasso do not achieve good recovery results, whereas the StructOMP algorithm achieves near perfect recovery of the original signal. We also include group Lasso in this experiment for illustration. We use predefined consecutive groups that do not completely overlap with the support of the signal. Since we do not know the correct group size, we just try group Lasso with several different group sizes (gs=2, 4, 8, 16). Although the results obtained with group Lasso are better than those of OMP and Lasso, they are still inferior to the results with StructOMP. As mentioned, this is because the pre-defined groups do not completely overlap with the support of the signal, which reduces the efficiency. In StructOMP, the base blocks are simply small connected line segments of size gs=3: that is, one node plus its two neighbors. This choice is only for simplicity, and it already produces good results in our experiments. If we include larger line segments into the base blocks (e.g., segments of size gs=4,5, etc), one can expect even better performance from StructOMP.

To study how the sample size $n$ affects the recovery performance, we vary the sample size and record the recovery results by different algorithms. To reduce the randomness, we perform the experiment 100 times for each sample size. Figure 3.3(a) shows the recovery performance in terms of Recovery Error and Sample Size, averaged over 100 random runs for each sample size. As expected, StructOMP is better than the group Lasso and far better than the OMP and Lasso. The results show that the proposed StructOMP can achieve better recovery performance for structured sparsity signals with less samples. Figure 3.3(b) shows the recovery performance in terms of CPU Time and Sample Size, averaged over 100 random runs for each sample size. The computation complexities of StructOMP and OMP are far lower than those of Lasso and Group Lasso.

It is worth noting that the performance of StructOMP is less stable than the other algorithms when the sample size $n$ is small. This is because for randomly generated design matrix, the structured RIP condition is only satisfied probabilistically. For small $n$, the necessary structured
RIP condition can be violated with relatively large probability, and in such case StructOMP does not have much advantage (at least theoretically). This implies the relatively large variance. The effect is much less noticeable with weakly sparse signal in Figure 3.7(a) because the necessary structured RIP condition is easier to satisfied for weakly sparse signals (based on our theory). Therefore the experimental results are consistent with our theory.

To study how the additive noise affects the recovery performance, we adjust the noise power $\sigma$ and then record the recovery results by different algorithms. In this case, the measurement number is $3k = 192$ for a fair comparison. To reduce the randomness, we also execute the experiment 100 times for each of noise level in performing each of algorithms. Figure 3.4(a) shows the recovery performance in terms of Recovery Error and Noise Level, averaged over 100 random runs for each noise level. As expected, StructOMP is also better than the group Lasso and far better than the OMP and Lasso. Moreover, the results show that these recovery algorithms can achieve better recovery performance for structured sparsity signals with lower noise power. Figure 3.4(b) shows the recovery performance in terms of CPU Time and Noise Level, averaged over 100 random runs for each sample size. The computation complexities of StructOMP and OMP are also lower than those of Lasso and Group Lasso for each noise level.

To further study the performance of the StructOMP, we compare it with other methods related with structured sparsity, such as OverlapLasso [70], ModelCS [7] and so on. For fair comparisons, both OverlapLasso and ModelCS use the same structure prior as that used in StructOMP. As introduced above, in StructOMP, the base blocks are simply small connected line segments of size $g_s=3$, which include one node plus its two neighbors; In OverlapLasso, the groups are also short connected line segments of size $g_s=3$; In ModelCS, the model gives the basic assumption: if one node is nonzero, then its two neighbors incline to be nonzeros with high probability. Figure 3.5(a) shows the recovery performance in terms of Recovery Error and Sample Size, averaged over 100 random runs for each sample size. As expected, StructOMP is better than OverlapLasso and ModelCS. The results show that the proposed StructOMP can achieve best recovery performance for structured sparsity signals with less samples. Figure 3.5(b) shows the recovery performance in terms of CPU Time and Sample Size, averaged over 100 random runs for each sample size. The computation complexity of StructOMP is lower than that of ModelCS (half) and are far lower than that of OverlapLasso. One of the reasons is that the OverlapLasso [70] requires to explicitly duplicate the variables in the design matrix and then solve the problem with the NonOverlap Group Lasso method. Therefore, it inevitably has higher computation complexity and requires more memory storages.
Figure 3.4: Recovery performance in terms of Noise Levels: (a) Recovery Error vs. Noise Level; (b) CPU Time vs. Noise Level.
Figure 3.5: Performance Comparisons between methods related with structured sparsity (ModelCS [7], OverlapLasso [70], StructOMP). (a) Recovery Error vs. Sample Size Ratio ($n/k$); (b) CPU Time vs. Sample Size Ratio ($n/k$).
Note that Lasso performs better than OMP in the first example. This is because the signal is strongly sparse (that is, all nonzero coefficients are significantly different from zero). In the second experiment, we randomly generate a 1D structured sparse signal with weak sparsity, where the nonzero coefficients decay gradually to zero, but there is no clear cutoff. One instance of a generated signal is shown in Figure 3.6 (a). Here, $p = 512$ and all coefficients of the signal are not zeros. We define the sparsity $k$ as the number of coefficients that contain 95% of the image energy. The support set of these signals is composed of $g = 2$ connected regions. Again, each element of the sparse coefficient is connected to two of its adjacent elements, which forms the underlying 1D line graph structure. The graph sparsity concept introduced earlier is used to compute the coding length of sparsity patterns in StructOMP. The projection matrix $X$ is generated by creating an $n \times p$ matrix with i.i.d. drawn from a standard Gaussian distribution $N(0, 1)$. For simplicity, the rows of $X$ are normalized to unit magnitude. Zero-mean Gaussian noise with standard deviation $\sigma = 0.01$ is added to the measurements.

Figure 3.6 shows one generated signal and its recovered results by different algorithms when $k = 32$ and $n = 48$. Again, we observe that OMP and Lasso do not achieve good recovery results, whereas the StructOMP algorithm achieves near perfect recovery of the original signal. As we do not know the predefined groups for group Lasso, we just try group Lasso with several different group sizes ($gs=2, 4, 8, 16$). Although the results obtained with group Lasso are better than those of OMP and Lasso, they are still inferior to the results with StructOMP. In order to study how the sample size $n$ affects the recovery performance, we vary the sample size and record the recovery results by different algorithms. To reduce the randomness, we perform the experiment 100 times for each of the sample sizes.

Figure 3.7(a) shows the recovery performance in terms of Recovery Error and Sample Size, averaged over 100 random runs for each sample size. As expected, StructOMP algorithm is superior in all cases. What’s different from the first experiment is that the recovery error of OMP becomes smaller than that of Lasso. This result is consistent with our theory, which predicts that if the underlying signal is weakly sparse, then the relatively performance of OMP becomes comparable to Lasso. Figure 3.7(b) shows the recovery performance in terms of CPU Time and Sample Size, averaged over 100 random runs for each sample size. The computation complexities of StructOMP and OMP are far lower than those of Lasso and Group Lasso.
Figure 3.6: Recovery results of 1D weakly sparse signal with line-structured sparsity. (a) original data; (b) recovered results with OMP (error is 0.5599); (c) recovered results with Lasso (error is 0.6686); (d) recovered results with Group Lasso (error is 0.4732 with group size gs=2); (e) recovered results with Group Lasso (error is 0.2893 with group size gs=4); (f) recovered results with Group Lasso (error is 0.2646 with group size gs=8); (g) recovered results with Group Lasso (error is 0.5459 with group size gs=16); (h) recovered results with StructOMP (error is 0.0846).

### 3.7.2 2D Image Compressive Sensing with Tree-structured Sparsity

It is well known that 2D natural images are sparse in a wavelet basis. Their wavelet coefficients have a hierarchical tree structure, which is widely used for wavelet-based compression algorithms [113]. Figure 3.8(a) shows a widely used example image with size 64 × 64: cameraman. Note that we use a reduced-size image instead of the original one for computational efficiency since the experiments is run many times with different random matrices. This reduction should not affect the relative performance among various algorithms.

In this experiment, each 2D wavelet coefficient of this image is connected to its parent coefficient and child coefficients, which forms the underlying hierarchical tree structure (which is a special case of graph sparsity). In our experiment, we choose Haar-wavelet to obtain its tree-structured sparsity wavelet coefficients. The projection matrix $X$ and noises are generated with the same method as that for 1D structured sparsity signals. OMP, Lasso and StructOMP are used to recover the wavelet coefficients from the random projection samples respectively. Then, the inverse wavelet transform is used to reconstruct the images with these recovered wavelet coefficients. Our task is to compare the recovery performance of the StructOMP to those of OMP and Lasso under the framework of compressive sensing.
Figure 3.7: Recovery performance for 1D Weak Line-Sparsity: (a) Recovery Error vs. Sample Size Ratio \((n/k)\); (b) CPU Time vs. Sample Size Ratio \((n/k)\).
For Lasso, we use identical regularization parameter for all coefficients (without varying regularization parameters based on bands or tree depth). For StructOMP, a simple block-structure is used, where each block corresponds to a node in the tree. This corresponds to setting $\delta = 0$ in Proposition 4.2, and thus suboptimal according to our theory. Nevertheless, we use this block set for efficiency only because the number of blocks is only linear in $p$. If we can show that this simple choice work well, then more complicated (although computationally less efficient) block set choices should lead to even better performance.

Figure 3.8 shows one example of the recovered results by different algorithms with sparsity number $k = 1133$ and sample size $n = 2048$. It shows that StructOMP obtains the best recovered result. Figure 3.9(a) shows the recovery performance in terms of Sample Size and Recovery Error, averaged over 100 random runs for each sample size. The StructOMP algorithm is better than both Lasso and OMP in this case. Since real image data are weakly sparse, the performance of standard OMP (without structured sparsity) is similar to that of Lasso. Figure 3.9(b) shows the recovery performance in terms of Sample Size and CPU Time, averaged over 100 random runs for each sample size. The computation complexity of StructOMP is compared with that of OMP and far lower than that of Lasso.

![Figure 3.8](image)

Figure 3.8: Recovery results with sample size $n = 2048$: (a) cameraman image; (b) recovered image with OMP (error is 0.1886; CPU time is 46.16s); (c) recovered image with Lasso (error is 0.1670; CPU time is 60.26s) and (d) recovered image with StructOMP (error is 0.0375; CPU time is 48.99s).

### 3.7.3 Background Subtracted Images for Robust Surveillance

Background subtracted images are typical structure sparsity data in static video surveillance applications. They generally correspond to the foreground objects of interest. Unlike the whole scene, these images are not only spatially sparse but also inclined to cluster into groups, which correspond to different foreground objects. Thus, the StructOMP algorithm can obtain superior recovery from compressive sensing measurements that are received by a centralized server...
Figure 3.9: Recovery performance for 2D wavelet tree sparsity: (a) Recovery Error vs. Sample Size; (b) CPU Time vs. Sample size.
from multiple and randomly placed optical sensors. In this experiment, the testing video is downloaded from http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. The background subtracted images are obtained with the software [157]. One sample image frame is shown in Figure 3.10(a). The support set of 2D images is thus composed of several connected regions. Here, each pixel of the 2D background subtracted image is connected to four of its adjacent pixels, forming the underlying graph structure in graph sparsity. We randomly choose 100 background subtracted images as test images.

Note that the color image have 3 channels. We can simply consider 3 channels independently and perform the sparse recovery algorithms in different channel respectively. On the other hand, in this application, 3 channels of the color background subtracted image share the same support set. There exists group sparsity in 3 color channels. Thus, a pixel in the color image can be considered as a triplet with three color intensities. In the comparisons, we can use the Group OMP and Group Lasso instead of OMP and Lasso, respectively.

Figure 3.10: Recovery results with sample size $n = 900$: (a) the background subtracted image; (b) recovered image with OMP (error is 1.1833); (c) recovered image with Lasso (error is 0.7075) and (d) recovered image with StructOMP (error is 0.1203).

In this experiment, we firstly consider the 3 color channel independently. Therefore, we use OMP, Lasso and StructOMP to recover each channel independently. The results shown in Figure 3.10 demonstrates that the StructOMP outperforms both OMP and Lasso in recovery. Figure 3.11(a) shows the recovery performance as a function of increasing sample size ratios. It demonstrates again that StructOMP significantly outperforms OMP and Lasso in recovery performance on video data. Comparing to the image compression example in the previous section, the background subtracted images have a more clearly defined sparsity pattern where nonzero coefficients are generally distinct from zero (that is, stronger sparsity); this explains why Lasso performs better than the OMP on this particular data. The results is again consistent with our theory. Figure 3.13(b) shows the recovery performance in terms of Sample Size and CPU Time, averaged over 100 random runs for each sample size. The computation complexity
Figure 3.11: Recovery performance. (a) Recovery Error vs. Sample Size; (b) CPU Time vs. Sample size.
of StructOMP is again compared with that of OMP and is far lower than that of Lasso.

If we consider a pixel as a triplet in the background subtracted image, we need to use Group OMP, Group Lasso and StructOMP to recover the color background subtracted images respectively. The results shown in Figure 3.12 demonstrates that the StructOMP outperforms both Group OMP and Group Lasso in recovery. Figure 3.13(a) shows the recovery performance as a function of increasing sample size ratios. It demonstrates again that StructOMP outperforms Group OMP and Group Lasso in this application. Figure 3.13(b) shows the recovery performance in terms of Sample Size and CPU Time, averaged over 100 random runs for each sample size. The computation complexity of StructOMP is again compared with that of Group OMP and is far lower than that of Group Lasso.

Figure 3.12: Recovery results with sample size $n = 600$: (a) the background subtracted image; (b) recovered image with Group OMP (error is 1.1340); (c) recovered image with Group Lasso (error is 0.6972) and (d) recovered image with StructOMP (error is 0.0808).

### 3.8 Discussion

This chapter develops a theory for structured sparsity where prior knowledge allows us to prefer certain sparsity patterns to others. Some examples are presented to illustrate the concept. The general framework established in this chapter includes the recently popularized group sparsity idea has a special case.

In structured sparsity, the complexity of learning is measured by the coding complexity $c(\hat{\beta}) \leq \|\hat{\beta}\|_0 + c(l(supp(\hat{\beta})))$ instead of $\|\hat{\beta}\|_0 \ln p$ which determines the complexity in standard sparsity. Using this notation, a theory parallel to that of the standard sparsity is developed. The theory shows that if the coding length $c(l(supp(\hat{\beta})))$ is small for a target coefficient vector $\hat{\beta}$, then the complexity of learning $\hat{\beta}$ can be significantly smaller than the corresponding complexity in standard sparsity. Experimental results demonstrate that significant improvements can be obtained on some real problems that have natural structures.

The structured greedy algorithm presented here is an efficient algorithm proposed to handle
Figure 3.13: Recovery performance: (a) Recovery Error vs. Sample Size; (b) CPU Time vs. Sample size.
the general structured sparsity learning. It is shown that the algorithm is effective under appropriate conditions. Future work include additional computationally efficient methods such as convex relaxation methods (e.g. $L_1$ regularization for standard sparsity, and group Lasso for strong group sparsity) and backward greedy strategies to improve the forward greedy method considered in this chapter.

3.9 Appendix: Proofs of Theorems

3.9.1 Proof of Proposition

Proposition 3.9.1 Let $P \in \mathbb{R}^{n \times n}$ be a projection matrix of rank $k$, and $y$ satisfies Assumption 2.5.1. Then there exists absolute constants $a_0$ and $b_0$ such that for all $\eta \in (0, 1)$, with probability larger than $1 - \eta$:

$$\|P(y - \mathbb{E}y)\|_2^2 \leq \sigma^2[7.4k + 2.7\sqrt{\ln(2/\eta)}].$$

Lemma 3.9.1 Consider a fixed vector $x \in \mathbb{R}^n$, and a random vector $y \in \mathbb{R}^n$ with independent sub-Gaussian components: $\mathbb{E}e^{t(y_i - \mathbb{E}y_i)} \leq e^{\sigma^2 t^2/2}$ for all $t$ and $i$, then $\forall \epsilon > 0$:

$$\Pr(|x^\top y - \mathbb{E}x^\top y| \geq \epsilon) \leq 2e^{-\epsilon^2/(2\sigma^2\|x\|_2^2)}.$$

Proof Let $s_n = \sum_{i=1}^n (x_i y_i - \mathbb{E}x_i y_i)$; then by assumption, $\mathbb{E}(e^{ts_n} + e^{-ts_n}) \leq 2\epsilon \sum_i x_i^2 \sigma^2 t^2/2$, which implies that $\Pr(|s_n| \geq \epsilon) e^{t^2} \leq 2\epsilon \sum_i x_i^2 \sigma^2 t^2/2$. Now let $t = \epsilon/\left(\sum_i x_i^2 \sigma^2\right)$, we obtain the desired bound.

The following lemma is taken from [104]. Since the proof is simple, it is included for completeness.

Lemma 3.9.2 Consider the unit sphere $S^{k-1} = \{x : \|x\|_2 = 1\}$ in $\mathbb{R}^k$ ($k \geq 1$). Given any $\epsilon > 0$, there exists an $\epsilon$-cover $Q \subset S^{k-1}$ such that $\min_{q \in Q} \|x - q\|_2 \leq \epsilon$ for all $\|x\|_2 = 1$, with $|Q| \leq (1 + 2/\epsilon)^k$.

Proof Let $B^k = \{x : \|x\|_2 \leq 1\}$ be the unit ball in $\mathbb{R}^k$. Let $Q = \{q_i\}_{i=1,\ldots,|Q|} \subset S^{k-1}$ be a maximal subset such that $\|q_i - q_j\|_2 > \epsilon$ for all $i \neq j$. By maximality, $Q$ is an $\epsilon$-cover of $S^{k-1}$. Since the balls $q_i + (\epsilon/2)B^k$ are disjoint and belong to $(1 + \epsilon/2)B^k$, we have

$$\sum_{i \leq |Q|} \text{vol}(q_i + (\epsilon/2)B^k) \leq \text{vol}((1 + \epsilon/2)B^k).$$
Therefore,

$$|Q|(\varepsilon/2)^k\text{vol}(B^k) \leq (1 + \varepsilon/2)^k\text{vol}(B^k),$$

which implies that $$|Q| \leq (1 + 2/\varepsilon)^k.$$ 

\[ \square \]

**Proof of Proposition**

According to Lemma 3.9.2, given $$\epsilon_1 > 0$$, there exists a finite set $$Q = \{q_i\}$$ with $$|Q| \leq (1 + 2/\epsilon_1)^k$$ such that $$\|Pq_i\|_2 = 1$$ for all $$i$$, and $$\min_i \|P\beta - Pq_i\|_2 \leq \epsilon_1$$ for all $$\|P\beta\|_2 = 1$$.

For each $$i$$, Lemma 3.9.1 implies that for all $$\epsilon_2 > 0$$:

$$\Pr(\|q_i^\top P(y - E_y)\|_2 \geq \epsilon_2) \leq 2e^{-\epsilon_2^2/(2\sigma^2)}.$$

Taking union bound for all $$q_i \in Q$$, we obtain with probability exceeding $$1 - 2(1 + 2/\epsilon_1)^k e^{-\epsilon_2^2/2\sigma^2}$$:

$$\|q_i^\top P(y - E_y)\|_2 \leq \epsilon_2$$

for all $$i$$.

Let $$\beta = P(y - E_y)/\|P(y - E_y)\|_2$$, then there exists $$i$$ such that $$\|P\beta - Pq_i\|_2 \leq \epsilon_1$$. We have

$$\|P(y - E_y)\|_2 = \beta^\top (y - E_y)$$

$$\leq \|P\beta - Pq_i\|_2 \|P(y - E_y)\|_2 + |q_i^\top P(y - E_y)|$$

$$\leq \epsilon_1 \|P(y - E_y)\|_2 + \epsilon_2.$$

Therefore

$$\|P(y - E_y)\|_2 \leq \epsilon_2/(1 - \epsilon_1).$$

Let $$\epsilon_1 = 2/15$$, and $$\eta = 2(1 + 2/\epsilon_1)^k e^{-\epsilon_2^2/2\sigma^2}$$, we have

$$\epsilon_2^2 = 2\sigma^2[(4k + 1) \ln 2 - \ln \eta],$$

and thus

$$\|P(y - E_y)\|_2 \leq \frac{15}{13} \sigma \sqrt{2(4k + 1) \ln 2 - 2 \ln \eta}.$$

This simplifies to the desired bound.

### 3.9.2 Proof of Theorem 2.5.1

We use the following lemma from [65].
Lemma 3.9.3 Suppose $X$ is generated according to Theorem 5.1. For any fixed set $F \subset \mathcal{I}$ with $|F| = k$ and $0 < \delta < 1$, we have with probability exceeding $1 - 3(1 + 8/\delta)ke^{-n\delta^2/8}$:

$$
(1 - \delta)\|\beta\|_2 \leq \frac{1}{\sqrt{n}}\|X_F\beta\|_2 \leq (1 + \delta)\|\beta\|_2 \quad (3.4)
$$

for all $\beta \in \mathbb{R}^k$.

Proof of Theorem 2.5.1

Since $\text{cl}(F)$ is a coding length, we have

$$
\sum_{F:|F|+\text{cl}(F)\leq s} (1 + 8/\delta)^{|F|} \leq \sum_{F:|F|+\gamma\text{cl}(F)\leq s} (1 + 8/\delta)^{|F|}
$$

$$
\leq \sum_{F} (1 + 8/\delta)^{|F|-\gamma\text{cl}(F)} = (1 + 8/\delta)^s \sum_{F} 2^{-\text{cl}(F)} \leq (1 + 8/\delta)^s,
$$

where we let $\gamma = 1/\log_2(1 + 8/\delta)$ in the above derivation.

For each $F$, we know from 3.9.3 that for all $\beta$ such that $\text{supp}(\beta) \subset F$:

$$
(1 - \delta)\|\beta\|_2 \leq \|X\beta\|_2 \leq (1 + \delta)\|\beta\|_2
$$

with probability exceeding $1 - 3(1 + 8/\delta)^{|F|}e^{-n\delta^2/8}$.

We can thus take the union bound over $F: |F|+\text{cl}(F) \leq s$, which shows that with probability exceeding

$$
1 - \sum_{F:|F|+\text{cl}(F)\leq s} 3(1 + 8/\delta)^{|F|}e^{-n\delta^2/8}
$$

the structured RIP in Equation (3.3) holds. Since

$$
\sum_{F:|F|+\text{cl}(F)\leq s} 3(1 + 8/\delta)^{|F|}e^{-n\delta^2/8} \leq 3(1 + 8/\delta)^se^{-n\delta^2/8} \leq e^{-t},
$$

we obtain the desired bound.

3.9.3 Proof of Theorem 2.5.2

Lemma 3.9.4 Suppose that Assumption 2.5.1 is valid. For any fixed subset $F \subset \mathcal{I}$, we have with probability $1 - \eta$, $\forall \beta$ such that $\text{supp}(\beta) \subset F$, and $a > 0$, we have

$$
\|X\beta - \mathbb{E}y\|^2_2 \leq (1 + a)(\|X\beta - y\|^2_2 - \|y - \mathbb{E}y\|^2_2) + (2 + a + a^{-1})\sigma^2[7.4|F| + 4.7\ln(4/\eta)].
$$

Proof Let

$$
P_F = X_F(X_F^\top X_F)^{-1}X_F^\top
$$
be projection matrix to the subspaces generated by columns of $X_F$.

Let $\tilde{a} = (I - P_F)E_y / \|(I - P_F)E_y\|_2$, $\delta_1 = \|P_F(y - E_y)\|_2$ and $\delta_2 = \|\tilde{a}^\top(y - E_y)\|_2$, we have

$$
\|X_\beta - E_y\|_2^2 \\
= \|X_\beta - y\|_2^2 - \|y - E_y\|_2^2 + 2(y - E_y)^\top (X_\beta - E_y)
$$

$\leq \|X_\beta - y\|_2^2 - \|y - E_y\|_2^2 + 2(y - E_y)^\top (X_\beta - P_F E_y) - 2\tilde{a}^\top (y - E_y)\|(I - P_F)E_y\|_2
$

$\leq \|X_\beta - y\|_2^2 - \|y - E_y\|_2^2 + 2\delta_1\|X_\beta - P_F E_y\|_2 + 2\delta_2\|(I - P_F)E_y\|_2
$

$\leq \|X_\beta - y\|_2^2 - \|y - E_y\|_2^2 + 2\sqrt{\delta_1^2 + \delta_2^2}\|X_\beta - P_F E_y\|_2 + \|(I - P_F)E_y\|_2
$

$\leq \|X_\beta - y\|_2^2 - \|y - E_y\|_2^2 + 2\sqrt{\delta_1^2 + \delta_2^2}\|X_\beta - E_y\|_2.
$

Note that in the above derivation, we have used the fact that $P_F X_\beta = X_\beta$, and $\|X_\beta - P_F E_y\|_2^2 + \|(I - P_F)E_y\|_2^2 = \|X_\beta - E_y\|_2^2$.

Now, by solving the above inequality, we obtain

$$
\|X_\beta - E_y\|_2^2 \leq \left[\sqrt{\|X_\beta - y\|_2^2 - \|y - E_y\|_2^2} + \delta_1 + \delta_2 + \sqrt{\delta_1^2 + \delta_2^2}\right]^2
$$

$$
\leq (1 + a)[\|X_\beta - y\|_2^2 - \|y - E_y\|_2^2] + (2 + a + 1/a)(\delta_1^2 + \delta_2^2).
$$

The desired bound now follows easily from Proposition 3.9.1 and Lemma 3.9.1, where we know that with probability $1 - \eta/2$,

$$
\delta_1^2 = (y - E_y)^\top P_F (y - E_y) \leq \sigma^2(7.4|F| + 2.7 \ln(4/\eta))
$$

and with probability $1 - \eta/2$,

$$
\delta_2^2 = |\tilde{a}^\top(y - E_y)|^2 \leq 2\sigma^2 \ln(4/\eta).
$$

We obtain the desired result by substituting the above two estimates and simplify.

\[\blacksquare\]

**Lemma 3.9.5** Suppose that Assumption 2.5.1 is valid. Then we have with probability $1 - \eta$, $\forall \beta \in \mathbb{R}^p$ and $a > 0$:

$$
\|X_\beta - E_y\|_2^2 \leq (1 + a) \left[\|X_\beta - y\|_2^2 - \|y - E_y\|_2^2\right] + (2 + a + 1/a)\sigma^2[7.4|F| + 4.7\ln(4/\eta)]
$$

**Proof** Note that for each $F$, with probability $2^{-\text{cl}(F)\eta}$, we obtain from Lemma 3.9.4 that $\forall \text{supp}(\beta) \in F$,

$$
\|X_\beta - E_y\|_2^2 \leq (1 + a) \left[\|X_\beta - y\|_2^2 - \|y - E_y\|_2^2\right] + (2 + a + 1/a)\sigma^2[7.4(|F| + \text{cl}(F)) + 4.7\ln(4/\eta)].
$$
Since $\sum_{F \subset \mathcal{I}, F \neq \emptyset} 2^{-\alpha(F)} \eta \leq \eta$, the result follows from the union bound.

**Lemma 3.9.6** Consider a fixed subset $\hat{F} \subset \mathcal{I}$. Given any $\forall \eta \in (0, 1)$, we have with probability $1 - \eta$, $\forall a > 0$:

$$\|X\hat{\beta} - \mathbb{E}y\|_2^2 - \|y - \mathbb{E}y\|_2^2 \leq \|X\hat{\beta} - \mathbb{E}y\|_2^2 + 2\sigma\sqrt{2\ln(2/\eta)\|X\hat{\beta} - \mathbb{E}y\|_2^2}.$$

**Proof** Let $\tilde{a} = (X\hat{\beta} - \mathbb{E}y)/\|X\hat{\beta} - \mathbb{E}y\|_2$, we have

$$\|X\hat{\beta} - \mathbb{E}y\|_2^2 - \|y - \mathbb{E}y\|_2^2 = [-2(X\hat{\beta} - \mathbb{E}y)^\top(y - \mathbb{E}y) + \|X\hat{\beta} - \mathbb{E}y\|_2^2] \leq 2\|X\hat{\beta} - \mathbb{E}y\|_2\|\tilde{a}^\top(y - \mathbb{E}y)\| + \|\mathbb{E}y - X\hat{\beta}\|_2^2.$$

The desired result now follows from Lemma 3.9.1.

**Lemma 3.9.7** Suppose that Assumption 2.5.1 is valid. Consider any fixed target $\hat{\beta} \in \mathbb{R}^p$. Then with probability exceeding $1 - \eta$, for all $\lambda \geq 0, \epsilon \geq 0, \hat{\beta} \in \mathbb{R}^p$ such that: $\hat{Q}(\hat{\beta}) + \lambda c(\hat{\beta}) \leq \hat{Q}(\hat{\beta}) + \lambda c(\hat{\beta}) + \epsilon$, and for all $a > 0$, we have

$$\|X\hat{\beta} - \mathbb{E}y\|_2^2 \leq (1 + a)[\|X\hat{\beta} - \mathbb{E}y\|_2^2 + 2\sigma\sqrt{2\ln(6/\eta)\|X\hat{\beta} - \mathbb{E}y\|_2^2}] + (1 + a)\lambda c(\hat{\beta}) + a'c(\hat{\beta}) + b'\ln(6/\eta) + (1 + a)\epsilon,$$

where $a' = 7.4(2 + a + a^{-1})\sigma^2 - (1 + a)\lambda$ and $b' = 4.7\sigma^2(2 + a + a^{-1})$. Moreover, if the coding scheme $c(\cdot)$ is sub-additive, then

$$np_-(c(\hat{\beta}) + c(\hat{\beta}))\|\hat{\beta} - \beta\|_2^2 \leq 10\|X\beta - \mathbb{E}y\|_2^2 + 2.5\lambda c(\hat{\beta}) + (37\sigma^2 - 2.5\lambda)c(\hat{\beta}) + 29\sigma^2\ln(6/\eta) + 2.5\epsilon.$$

**Proof** We obtain from the union bound of Lemma 3.9.5 (with probability $1 - \eta/3$) and Lemma 3.9.6 (with probability $1 - 2\eta/3$) that with probability $1 - \eta$:

$$\|X\hat{\beta} - \mathbb{E}y\|_2^2 \leq (1 + a)\left[\|X\hat{\beta} - \mathbb{E}y\|_2^2 - \|y - \mathbb{E}y\|_2^2\right] + (2 + a + a^{-1})[7.4\sigma^2c(\hat{\beta}) + 4.7\sigma^2\ln(6/\eta)]$$

$$\leq (1 + a)[\|X\hat{\beta} - \mathbb{E}y\|_2^2 - \|y - \mathbb{E}y\|_2^2 + \lambda c(\hat{\beta}) + \epsilon] + a'c(\hat{\beta}) + b'\ln(6/\eta) + (1 + a)\epsilon$$

$$\leq (1 + a)[\|X\hat{\beta} - \mathbb{E}y\|_2^2 + 2\sigma\sqrt{2\ln(6/\eta)\|X\hat{\beta} - \mathbb{E}y\|_2^2}] + (1 + a)\lambda c(\hat{\beta}) + a'c(\hat{\beta})$$

$$+ b'\ln(6/\eta) + (1 + a)\epsilon$$
This proves the first claim of the theorem.

The first claim with $a = 1$ implies that

\[
\|X\hat{\beta} - X\tilde{\beta}\|_2^2 \leq \|X\hat{\beta} - \mathbb{E}y\|_2^2 + \|X\tilde{\beta} - \mathbb{E}y\|_2^2 + 7.5\|X\tilde{\beta} - \mathbb{E}y\|_2 + 2.5\lambda c(\tilde{\beta}) + 1.25(29.6\sigma^2 - 2\lambda)c(\hat{\beta}) \\
+ 1.25\|X\hat{\beta} - \mathbb{E}y\|_2 + 5\|X\tilde{\beta} - \mathbb{E}y\|_2 \\
\leq 7.5\|X\tilde{\beta} - \mathbb{E}y\|_2 + 5\sigma\sqrt{2\ln(6/\eta)}\|X\tilde{\beta} - \mathbb{E}y\|_2 + 2.5\lambda c(\tilde{\beta}) + 1.25(29.6\sigma^2 - 2\lambda)c(\hat{\beta}) \\
+ 1.25\times 18.8\sigma^2 \ln(6/\eta) + 2.5\epsilon \\
\leq 10\|X\tilde{\beta} - \mathbb{E}y\|_2^2 + 2.5\lambda c(\tilde{\beta}) + (37\sigma^2 - 2.5\lambda)c(\hat{\beta}) + 29\sigma^2 \ln(6/\eta) + 2.5\epsilon.
\]

Since $c(\hat{\beta} - \tilde{\beta}) \leq c(\beta) + c(\tilde{\beta}) \leq s + c(\tilde{\beta})$, we have $\|X\hat{\beta} - X\tilde{\beta}\|_2^2 \geq n\rho -(s + c(\tilde{\beta}))(\hat{\beta} - \tilde{\beta})^2$. This implies the second claim.

\[\blacksquare\]

**Proof of Theorem 2.5.2**

We take $\lambda = 0$ in Lemma 3.9.7, and obtain:

\[
\|X\hat{\beta} - \mathbb{E}y\|_2^2 \leq (1 + a)(\|X\tilde{\beta} - \mathbb{E}y\|_2^2 + 2\sigma\sqrt{2\ln(6/\eta)}\|X\tilde{\beta} - \mathbb{E}y\|_2 \\
+ 7.4(2 + a + a^{-1})\sigma^2 c(\tilde{\beta}) + 4.7\sigma^2 (2 + a + a^{-1}) \ln(6/\eta)) + (1 + a)\epsilon \\
= (\|X\tilde{\beta} - \mathbb{E}y\|_2 + \sigma\sqrt{2\ln(6/\eta)})^2 + 14.8\sigma^2 c(\tilde{\beta}) + 7.4\sigma^2 \ln(6/\eta) + \epsilon \\
+ a(\|X\tilde{\beta} - \mathbb{E}y\|_2 + \sigma\sqrt{2\ln(6/\eta)})^2 + 7.4\sigma^2 c(\tilde{\beta}) + 2.7\sigma^2 \ln(6/\eta) + \epsilon \\
+ a^{-1}[7.4\sigma^2 c(\tilde{\beta}) + 4.7\sigma^2 \ln(6/\eta)].
\]

Now let $z = \|X\tilde{\beta} - \mathbb{E}y\|_2 + \sigma\sqrt{2\ln(6/\eta)}$, and we choose $a$ to minimize the right hand side as:

\[
\|X\hat{\beta} - \mathbb{E}y\|_2^2 \leq z^2 + 14.8\sigma^2 c(\tilde{\beta}) + 7.4\sigma^2 \ln(6/\eta) + \epsilon \\
+ 2[z^2 + 7.4\sigma^2 c(\tilde{\beta}) + 2.7\sigma^2 \ln(6/\eta) + \epsilon]^{1/2}[7.4\sigma^2 c(\tilde{\beta}) + 4.7\sigma^2 \ln(6/\eta)]^{1/2} \\
\leq [(z^2 + 7.4\sigma^2 c(\tilde{\beta}) + 2.7\sigma^2 \ln(6/\eta) + \epsilon)^{1/2} + (7.4\sigma^2 c(\tilde{\beta}) + 4.7\sigma^2 \ln(6/\eta))^{1/2}]^2 \\
\leq [z + 2(7.4\sigma^2 c(\tilde{\beta}) + 2.7\sigma^2 \ln(6/\eta) + \epsilon)]^{1/2}.
\]

This proves the first inequality. The second inequality follows directly from Lemma 3.9.7 with $\lambda = 0$.

**3.9.4 Proof of Theorem 2.5.3**

The following lemma is an adaptation of a similar result in [152] on greedy algorithms for standard sparsity.
Lemma 3.9.8 Suppose the coding scheme is sub-additive. Consider any \( \beta \), and a cover of \( \beta \) by \( \mathcal{B} \):

\[
supp(\beta) \subset \bar{F} = \bigcup_{j=1}^b \bar{B}_j \quad (\bar{B}_j \in \mathcal{B}),
\]

Let \( c(\beta, \mathcal{B}) = \sum_{j=1}^b c(\bar{B}_j) \). Let \( \rho_0 = \max_j \rho(\bar{B}_j) \). Then for all \( \beta \) such that \( \text{supp}(\beta) \subset F \), \( c(\bar{B}_j \cup F) \geq c(F) \), and \( \|X\beta - y\|_2^2 \geq \|X\bar{\beta} - y\|_2^2 \), we have

\[
\max_j \phi(\bar{B}_j) \geq \frac{\rho_{\text{avg}}(F \cup F)}{\rho_0} \left[ \|X\beta - y\|_2^2 - \|X\bar{\beta} - y\|_2^2 \right],
\]

where \( \phi(B) \) is defined in (3).

Proof For all \( \ell \in F \), \( \|X\beta + \alpha Xe_\ell - y\|_2^2 \) achieves the minimum at \( \alpha = 0 \). This implies that

\[
x_\ell^T (X\beta - y) = 0
\]

for all \( \ell \in F \). Therefore we have

\[
(X\beta - y)^T \sum_{\ell \in F - F} (\bar{\beta}_\ell - \beta_\ell)x_\ell
= (X\beta - y)^T \sum_{\ell \in F \cup F} (\bar{\beta}_\ell - \beta_\ell)x_\ell = (X\beta - y)^T (X\bar{\beta} - X\beta)
= -\frac{1}{2}\|X(\bar{\beta} - \beta)\|_2^2 + \frac{1}{2}\|X\bar{\beta} - y\|_2^2 - \frac{1}{2}\|X\beta - y\|_2^2.
\]

Now, let \( \bar{B}_j' \subset \bar{B}_j - F \) be disjoint sets such that \( \bigcup_j \bar{B}_j' = \bar{F} - F \). The above inequality leads to the following derivation \( \forall \eta > 0 \):

\[
- \sum_j \phi(\bar{B}_j)(c(\bar{B}_j \cup F) - c(F))
\leq \sum_j \left( \|X\beta + \eta \sum_{\ell \in B_j'} (\bar{\beta}_\ell - \beta_\ell)x_\ell - y\|_2^2 - \|X\beta - y\|_2^2 \right)
\leq \eta^2 \sum_{\ell \in F - F} (\bar{\beta}_\ell - \beta_\ell)^2 \rho_0 + 2\eta\|X(\bar{\beta} - \beta)\|_2^2 \sum_{\ell \in F - F} (\bar{\beta}_\ell - \beta_\ell)x_\ell
\leq \eta^2 \sum_{\ell \in F - F} (\bar{\beta}_\ell - \beta_\ell)^2 \rho_0 - \eta\|X(\bar{\beta} - \beta)\|_2^2 + \eta\|X\bar{\beta} - y\|_2^2 - \eta\|X\beta - y\|_2^2.
\]

By optimizing over \( \eta \), we obtain

\[
\max_j \phi(\bar{B}_j) \sum_j c(\bar{B}_j) \geq \sum_j \phi(\bar{B}_j)(c(\bar{B}_j \cup F) - c(F))
\geq \frac{\|X(\bar{\beta} - \beta)\|_2^2 + \|X\beta - y\|_2^2 - \|X\bar{\beta} - y\|_2^2}{4 \sum_{\ell \in F - F} (\bar{\beta}_\ell - \beta_\ell)^2 \rho_0 + 2\eta\|X(\bar{\beta} - \beta)\|_2^2 + \eta\|X\bar{\beta} - y\|_2^2 - \eta\|X\beta - y\|_2^2}
\geq \frac{\rho_{\text{avg}}(F \cup F)}{\rho_0} \left[ \|X\beta - y\|_2^2 - \|X\bar{\beta} - y\|_2^2 \right].
\]
This leads to the desired bound.

**Proof of Theorem 2.5.3**

Let 

$$\gamma' = \frac{\gamma \rho \cdot (s + c(\bar{F}))}{\rho_0(\mathcal{B}) c(\beta, \mathcal{B})}.$$  

By Lemma 3.9.8, we have at any step $k > 0$:

$$\|X\beta^{(k-1)} - y\|_2^2 - \|X\beta^{(k)} - y\|_2^2 \geq \gamma' \left[ \|X\beta^{(k-1)} - y\|_2^2 - \|X\beta - y\|_2^2 \right] (c(\beta^{(k)}) - c(\beta^{(k-1)})),$$

which implies that

$$\max[0, \|X\beta^{(k)} - y\|_2^2 - \|X\beta - y\|_2^2] \leq \max[0, \|X\beta^{(k-1)} - y\|_2^2 - \|X\bar{\beta} - y\|_2^2] e^{-\gamma'c(\beta^{(k)}) - c(\beta^{(k-1)})}.$$  

Therefore at stopping, we have

$$\|X\beta^{(k)} - y\|_2^2 - \|X\bar{\beta} - y\|_2^2 \leq \left[ \|y\|_2^2 - \|X\bar{\beta} - y\|_2^2 \right] e^{-\gamma'c(\beta^{(k)}) - c(\beta^{(k-1)})},$$

This proves the theorem.

**3.9.5 Proof of Theorem 2.5.4**

Given $s'$, we consider $f_j = \min_{\ell \geq j} \hat{Q}(\beta(s/2^\ell))$. Then by Lemma 3.9.6, we have with probability $1 - 2^{-j-1} \eta$:

$$\left| f_j - \|y - E\beta y\|_2^2 \right| \leq 2an2^{\eta_1}/s^{p} + 2\sigma^2[j + 1 + \ln(2/\eta)].$$

Now, by taking $\epsilon = 2an/s^{p} + 2\sigma^2[\ln(2/\eta)]$ in Theorem 5.5, we obtain

$$\sum_{j=0}^{\infty} 2^{-j} \ln \frac{f_{j+1} - f_0 + \epsilon}{f_j - f_0 + \epsilon} \leq \sum_{j=0}^{\infty} 2^{-j} \ln(1 + (f_{j+1} - f_0)/\epsilon) \leq \sum_{j=0}^{\infty} 2^{-j} \ln(2 + j + 2^{p j}) \leq \sum_{j=0}^{\infty} 2^{-j}(2 + j + p j \ln 2) = 2 + 2(1 + p \ln 2),$$
where we have used the simple inequality \( \ln(\alpha + \beta) \leq \alpha + \ln(\beta) \) when \( \alpha \geq 2 \) and \( \beta \geq 1 \). Therefore,

\[
s \geq \frac{\rho_0(B)}{\gamma \rho_-(s + c(F))} s'(6 + 2p) \\
\geq \frac{\rho_0(B)}{\gamma \rho_-(s + c(F))} s' \left[ 1.4 + \sum_{j=0}^{\infty} 2^{-j} \ln \frac{f_{j+1} - f_0 + \epsilon}{f_j - f_0 + \epsilon} \right].
\]

This means that we can obtain the desired bound.
Chapter 4
Applications in Compressive Sensing and Computer Vision

This chapter introduces the applications of structured sparsity in compressive sensing and computer vision. A new learning formulation is introduced, which is called dynamic group sparsity. It is a special model of structured sparsity, and is motivated by the observation that in some practical sparse data the nonzero coefficients are often not random but tend to be clustered. Intuitively, better results can be achieved in these cases by reasonably utilizing both clustering and sparsity priors. Motivated by this idea, we have developed a new greedy sparse recovery algorithm, which prunes data residues in the iterative process according to both sparsity and group clustering priors rather than only sparsity as in previous methods. In compressive sensing, the proposed algorithm can recover stably sparse data with clustering trends using far fewer measurements and computations than current state-of-the-art algorithms with provable guarantees. Moreover, our algorithm can adaptively learn the dynamic group structure and the sparsity number if they are not available in the practical applications, such as video background tracking and visual tracking, etc.

4.1 Introduction

The compressive sensing (CS) theory has shown that a sparse signal can be recovered from a small number of its linear measurements with high probability [21, 32]. According to CS, a sparse signal $x \in \mathbb{R}^p$ should be recovered from the following linear random projections:

$$y = \Phi x + e,$$

where $y \in \mathbb{R}^n$ is the measurement vector, $\Phi \in \mathbb{R}^{n \times p}$ is the random projection matrix, $n \ll p$, and $e$ is the measurement noise. The CS theory is magnetic as it implies that the signal $x \in \mathbb{R}^p$ can be recovered from only $n = O(k \log(p/k))$ measurements [21] if $x$ is a $k$-sparse signal, which means that $x \in \mathbb{R}^p$ can be well approximated using $k \ll p$ nonzero coefficients under some linear transformations. It directly leads to the potential of cost saving in digital data capturing.
Although the encoding in data capturing only involves simple linear projections, signal recovery requires nonlinear algorithms to seek the sparsest signal from the measurements. This problem can be formulated with $L_0$ minimization:

$$x_0 = \text{argmin} \|x\|_0 \quad \text{while} \quad \|y - \Phi x\|^2 < \varepsilon$$

(4.2)

where $\|\cdot\|_0$ denotes the $L_0$-norm that counts the number of nonzero entries and $\varepsilon$ is the noise level. This problem is NP-hard. In the general case, no known procedure can correctly find the sparsest solution more efficiently than exhausting all subsets of the entries for $x$. One key problem in CS is thus to develop efficient recovery algorithms with nearly optimal theoretical performance guarantees.

However, in some practical applications, the nonzero sparse coefficients are often not randomly distributed but group-clustered. They tend to cluster into groups although these clustering group structures are dynamic and unpredictable. (For example, the group number/size/location may be unknown.) A few attempts have been made to utilize these group clustering priors for better sparse recovery [14, 72, 128, 121, 148]. For simplicity, all of them assume that the group structures (such as the group number/size/location) are known before recovery. Moreover, they only consider the case where all groups share a common nonzero coefficient support set $^1$. These recovery algorithms either do not have explicit bounds on the minimal number of measurements, or lack of provable recovery performance guarantees from noise measurements. While their assumption of the block sparsity structure enables better recovery from fewer measurements with less computation, it is not flexible enough to handle some practical sparse data in which the group structures are unknown and only the sparse group-clustering trend is known. Therefore, none of them can handle dynamic group clustering priors, when we do not know the group structure, and only know the sparsity and group clustering trend.

In this chapter, a dynamic group sparsity recovery algorithm is then proposed based on the theory of structured sparsity. It assumes that the dynamic group clustering sparse signals lie in a union of subspaces [17, 82] and proposes an approximation algorithm in this union of subspaces to iteratively prune the signal estimations according to both sparsity and group clustering priors. The group clustering trend implies that, if a point lives in the union of subspaces, its neighboring points would also live in this union of subspaces with higher probability, and vice versa. By enforcing this constraints, the degrees of freedom of the sparse signals have been significantly reduced to a narrower union of subspaces. It leads to several advantages: 1) accelerating

---

$^1$The support set of sparse data $x$ is defined as the set of indices corresponding to the nonzero entries in $x$ and denoted by $\text{supp}(x)$.
the signal pruning process; 2) decreasing the minimal number of necessary measurements; and 3) improving robustness to noise and preventing the recovered data from having artifacts. These advantages enable the proposed algorithm to efficiently obtain stable sparse recovery with far fewer measurements than previous algorithms. Finally, we extended the proposed algorithm to adaptively learn the sparsity numbers when they are not exactly known in practical applications.

4.2 Theory Review

As we know, the decreasingly sorted coefficients of many real signals rapidly decay according to the power law. Thus, these signals can be well approximated or compressed to $k$-sparse signals although they are not strictly sparse. In CS, the signal capture and compression are integrated into a single process [20, 32]. Thus, we do not capture a sparse signal $x \in \mathbb{R}^p$ directly but rather capture $n < p$ linear measurements $y = \Phi x$ based on a measurement matrix $\Phi \in \mathbb{R}^{n \times p}$. Suppose the set of $k$-sparse signals $x \in \mathbb{R}^p$ lives in the union $\Omega_k$ of $k$-dimensional subspaces, the union $\Omega_k$ thus includes $C_k^p$ subspaces. To stably recover the $k$-sparse signal $x$ from $n$ measurements, the measurement matrix $\Phi$ is required to satisfy the Restricted Isometry Property (RIP) [20].

Definition: (k-RIP) A matrix $\Phi \in \mathbb{R}^{n \times p}$ is said to have $k$-restricted isometry property (k-RIP) with constant $\delta_k > 0$ if, for all $x$ in the union $\Omega_k$,

$$
(1 - \delta_k) \| x \|_2^2 \leq \| \Phi x \|_2^2 \leq (1 + \delta_k) \| x \|_2^2
$$

(4.3)

While the sparse signal $x$ lives in a union of subspaces $A \subset \mathbb{R}^p$, the k-RIP can be extended to the $A$-RIP [17]:

Definition: (A-RIP) A matrix $\Phi \in \mathbb{R}^{n \times p}$ is said to have $A$-restricted isometry property (A-RIP) with constant $\delta_A(\Phi)$ if, for all $x$ living in the union of subspaces $A$,

$$
(1 - \delta_A(\Phi)) \| x \|_2^2 \leq \| \Phi x \|_2^2 \leq (1 + \delta_A(\Phi)) \| x \|_2^2
$$

(4.4)

Blumensath and Davies have proposed one theorem on the sufficient condition for stable sparse recovery to guide the minimal measurement number $n$ necessary for a subgaussian random measurement matrix to have the $A$-RIP with the given probability [17]:

Lemma 1: Suppose $A_k \subset \mathbb{R}^p$ be the union of $L$ subspaces of $k$-dimensions aligned with $\mathbb{R}^p$. For any $t > 0$, if

$$
m \geq \frac{2}{c_0} (\log(2L) + k \log(\frac{12}{\delta_{A_k}}) + t)
$$

(4.5)
then the subgaussian random matrix $\Phi \in \mathbb{R}^{n \times p}$ has the $A$-RIP with constant $\delta_{A_k}$, where $0 < \delta_{A_k} < 1$, and $c > 0$ only depends on the $\delta_{A_k}$. The probability is at least $1 - e^{-t}$.

For the $k$-sparse data recovery, we know that $A_k \subset \mathbb{R}^p$ is the union of $L = C_k^p$ subspaces. Thus, this theorem directly leads to the classic CS result $n = O(k + k \log(p/k))$.

### 4.3 Dynamic Group Sparsity

The success of sparse recovery in compressive sensing motivates us to further observe the support set of the sparse coefficients. Observations on some practical sparse data show that the support sets often have the group clustering trend with dynamic and unknown group structure. Intuitively, the measurement number bound may be further reduced if this trend can be dexterously utilized as sparsity in convention CS. In this section, we propose a new algorithm to seamlessly combine this prior with sparsity, which is shown to enable better recovery results for this case with less measurement requirement and lower computation complexity.

#### 4.3.1 Dynamic Group Sparse Data

![Figure 4.1: An example of dynamic group sparse data: (a) one video frame; (b) the foreground image; (c) the foreground mask and (d) the background subtracted image with the proposed algorithm.](image)

Similar to the definition of $k$-sparse data, we can define dynamic group sparse data as follow:

**Definition:** ($G_{k,q}$-sparse data) A data $x \in \mathbb{R}^p$ is defined as the dynamic group sparse data ($G_{k,q}$-sparse data) if it can be well approximated using $k \ll p$ nonzero coefficients under some linear transforms and these $k$ nonzero coefficients are clustered into $q \in \{1, \cdots, k\}$ groups.

From this definition, we can know that $G_{k,q}$-sparse data only requires that the nonzero coefficients in the sparse data have the group clustering trend and does not require to know any information about the group size and location. In the following, it will be further illustrated that the group number $q$ is also not necessary to be known in our algorithm. The group structures can be dynamic and unknown. Figure 4.1 shows a real sample of $G_{k,q}$-sparse data in a video...
surveillance application. We can find that nonzero coefficients are not randomly distributed but clustered spatially in the background subtracted image (Figure 4.1 (b)) and the foreground mask (Figure 4.1 (c)). More specially, the R, G and B channels of the background subtracted image share a common support set although the nonzero coefficients are spatially clustered in each channel respectively.

Due to the additional dynamic clustering prior, the union of subspaces containing \(G_{k,q}\)-sparse data does not span all \(k\)-dimensional subspaces of the union \(\Omega_k\) as in the conventional CS [21, 32, 29, 97, 127]. The former is far narrower than the latter in most cases. The dynamic group clustering prior significantly reduces the degrees of freedom of the sparse signal since it only permits certain combinations of its support set rather than all random combinations. This will make it possible for us to decrease the minimal measurement number \(m\) for stable recovery. Therefore, we need to develop a new theorem on the sufficient condition for stable dynamic group sparse data recovery.

**Lemma 2:** Suppose \(G_{k,q}\)-sparse data \(x \in \mathbb{R}^p\) is known to live in \(L_{k,q}\) \(k\)-dimensional subspaces aligned to \(\mathbb{R}^p\), and \(G_{k,q}\) is the smallest union of these \(k\)-dimension subspaces. For any \(t > 0\), if

\[
n = O(k + q \log(p/q))
\]

then the subgaussian random matrix \(\Phi \in \mathbb{R}^{n \times p}\) has the \(A\)-RIP with constant \(\delta_{A_k}\), where \(0 < \delta_{A_k} < 1\). The probability is at least \(1 - e^{-t}\).

**Proof:** It can be easily proved based on the theory Struct-RIP introduced in the previous chapter. In the following, another proof is given from a different view.

**Proof:** According to Lemma 1, the key point of the proof is to find the value or upper bound of \(L_{k,q}\). After we obtain its value or upper bound, we can directly obtain the upper bound of the measurement number \(m\) according to equation 4.5 in Lemma 1. Suppose the dimension of the sparse signal is \(d\). We know that \(k\) nonzero coefficients of \(G_{k,q}\)-sparse data \(x \in \mathbb{R}^p\) have clustered into \(q\) groups. Suppose that the \(i\)-th group has \(r_i\) nonzero coefficients. Thus, \(r_i, i = 1, \ldots, q\) should be a natural number and their sum is \(k\). Without loss of generality, we can assume that \(p\) coefficients of \(x \in \mathbb{R}^p\) are divided into \(q\) regions, where the \(i\)-th region has \(\frac{pr_i}{k}\) nonzero coefficients and every region is not overlapped with other regions (While \(p/k\) is not an integer, we can round them off and just keep their sum with \(p\)). Considering the restrictions that \(r_i, i = 1, \ldots, q\) should be a natural number and their sum is \(k\), there are \(\binom{q}{k-1}\) possible combinations. According to Stirling’s formula, \(\binom{q}{k-1} \leq e^\Theta(k/q)^{q-1}\). For each combination, we divide the original problem into \(q\) small problems. For each small problem, it is equal to the
case where all \( r_i \) nonzero coefficients are clustered into one group. Thus, \( L_{r_i,1} \leq \frac{pr_i}{k} (2d - 1)^{r_i - 1} \) for each small problem. Then, the number of all possible combinations for the original problem is:

\[
L_{k,q} \leq \left( \prod_{i=1}^{q} \frac{pr_i}{k} \right)(2d - 1)^{r_i - 1} C_{k-1}^{q-1} \\
\leq \left( \frac{2d}{q} \right)^q (2d - 1)^k - q^3 (k/q)^{q-1}
\]

(4.7)

If we do consider the overlapping problems between the nonzero coefficients of neighboring regions, the number of all allowed combinations for \( d \)-dimensional \( G_{k,q} \)-sparse data should be further less than the right of this equation. But the bound in equation 4.7 is enough for our proof. According to equation 4.5 in Lemma 1, we know:

\[
n \geq \frac{2}{\epsilon_0 A_k} (\log(2L_{k,q}) + k \log(\frac{12}{\delta A_k}) + t)
\]

(4.8)

Thus, we can easily obtain \( n = O(k + q\log(p/q)) \), which proves the Lemma.

Lemma 2 shows that the number of measurements required for robustly recovering dynamic group sparsity data is \( m = O(k + q\log(p/q)) \), which is a significant improvement over the \( n = O(k + k\log(p/k)) \) that would be required by conventional CS recovery algorithms [21, 32, 97, 127]. While the group number \( q \) is smaller, more improvements can be obtained. While \( q \) is far smaller than \( k \) and \( k \) is close to \( \log(p) \), we can get \( n = O(k) \). Note that, this is a sufficient condition. If we know more priors about group settings, we can further reduce this bound.

### 4.3.2 Dynamic Group Sparsity Recovery

Lemma 2 equips us to propose a new recovery algorithm for dynamic group sparse data, namely dynamic group sparsity (DGS) recovery algorithm. From the introduction, we know that only the SP [29] and the CoSaMP [97] have better balance between theoretical guarantee and computation complexity among existing greedy recovery algorithms. Actually, these two algorithms have a similar framework. In this section, we demonstrate how to seamlessly integrate the dynamic group clustering prior into that framework.

Figure 4.2 shows the flow chart of our proposed recovery algorithm. There are five main steps in each iteration: 1) pruning the residue estimation; 2) merging the support sets; 3) estimating the signal by least square; 4) pruning the signal estimation and 5) updating the signal/residue estimation and support set. One can observe that it is similar to that of SP/CoSaMP algorithms. The difference only exists in the pruning process in step 1 and step 4. The modification is simple. We prune the estimation in the step 1 and step 4 using DGS approximation pruning rather than \( k \)-sparse approximation, as we only need to search over subspaces of \( A_{k,q} \) instead of \( C_p^k \) subspaces of \( \Omega_k \). It directly leads to fewer measurement requirement for stable data recovery.
Figure 4.2: Main steps of the proposed DGS Recovery algorithm
Algorithm 2 DGS approximation pruning

**Input:** \( x \in \mathbb{R}^p \) (estimations); \( k \) (the sparsity number); \( J_y \) (task number); \( J_b \) (block size); 
\( N_x \in \mathbb{R}^{p \times \tau} \) (values of x’s neighbors); \( w \in \mathbb{R}^{p \times \tau} \) (weights for neighbors); \( \tau \) (neighbor number)

\[
J_x = J_y J_b; x \in \mathbb{R}^p \text{ is shaped to } x \in \mathbb{R}^{\frac{J_x}{J_y} \times J_x}
\]

\[
N_x \in \mathbb{R}^{p \times \tau} \text{ is shaped to } N_x \in \mathbb{R}^{\frac{J_x}{J_y} \times J_x \times \tau};
\]

**for all** \( i = 1, \ldots, \frac{J_x}{J_y} \) **do**

- Combining each entry with its neighbors:

\[
z(i) = \sum_{j=1}^{J_y} x^2(i, j) + \sum_{j=1}^{J_y} \sum_{t=1}^{\tau} w^2(i, t) N_x^2(i, j, t)
\]

**end for**

\( \Omega \in \mathbb{R}^{\frac{J_x}{J_y} \times 1} \) is set as indices corresponding to the largest \( k/J_x \) entries of \( z \)

**for all** \( j = 1, \ldots, J_x \) **do**

- **for all** \( i = 1, \ldots, \frac{J_x}{J_y} \) **do**

- Obtain the final list:

\[
\Gamma((j - 1) \frac{k}{J_x} + i) = (j - 1) \frac{k}{J_x} + \Omega(i)
\]

**end for**

**end for**

**Output:** \( supp(x, k) \leftarrow \Gamma \)

The DGS pruning algorithm is described in algorithm 2. There exist two prior-dependent parameters \( J_y \) and \( J_b \). \( J_y \) is the number of tasks if the problem can be represented as a multi-task CS problem [72]. \( J_b \) is the block size if the interested problem can be modelled as a block sparsity problem [14, 128, 121, 148]. Their default values are set as 1, which is the case of traditional sparse recovery in compressive sensing. Moreover, there are two important user-tuning parameters, the weight \( w \) of neighbors and the neighbor number \( \tau \) of each element in sparse data. In practice, it is very straightforward to adjust them since they have the physical meanings. The first one controls the balance between the sparsity prior and the group clustering prior. While \( w \) is smaller/bigger, it means that the degree of dynamic group clustering is lower/higher in the sparse signal. Generally, they are set as 0.5’s if there are no more knowledge about that in practice. The parameter \( \tau \) controls the number of neighbors that can be affected by each element in sparse data. Generally, it is good enough to set it as 2, 4 and 6 for 1D, 2D and 3D data respectively.

Up to now, we assume that we know the sparsity number \( k \) of the sparse data before recovery. However, it is not always true in practical applications. For example, we do not know the exact sparsity numbers of the background subtracted images although we know they tend to be dynamic group sparse. Motivated by the idea in [31], we develop a new recovery algorithm called AdaDGS by incorporating an adaptive sparsity scheme into the above DGS
Algorithm 3 AdaDGS Recovery

1: Input: $\Phi \in \mathbb{R}^{m \times n}$ \{sample matrix\}; $y \in \mathbb{R}^m$ \{sample vector\}; $[k_{\text{min}}, k_{\text{max}}]$ \{sparsity range\}; $\Delta k$ \{sparsity step size\}
2: Initialization: residue $y_r = y$; $\Gamma = \text{supp}(x) = \emptyset$; sparse data $x = 0$; sparsity number $k = k_{\text{min}}$
3: repeat
4: Perform DGS recovery algorithm with sparsity number $k$ to obtain $x$ and the residue
5: if halting criterion false then
6: Update $\Gamma$, $y_r$ and $k = k + \Delta k$
7: end if
8: until halting criterion true
9: Output: $x = \Phi^\dagger y$

recovery algorithm.

Suppose the range of the sparsity number is known to be $[k_{\text{min}}, k_{\text{max}}]$. We can set the step size of sparsity number as $\Delta k$. The whole recovery process is divided into several stages, each of which includes several iterations. Thus, there are two loops in AdaDGS recovery algorithm. The sparsity number is initialized as $k_{\text{min}}$ before iterations. During each stage (inner loop), we iteratively optimize sparse data with the fixed sparsity number $k_{\text{curr}}$ until the halting condition within the stage is true (for example, the residual norm is not decreasing). We then switch to the next stage after adding $\Delta k$ into the current sparsity number $k_{\text{curr}}$ (outer loop). The whole iterative process will stop whenever the halting condition is satisfied. For practical applications, there is a trade-off between the sparsity step size $\Delta k$ and the recovery performance. Smaller step sizes require more iterations and bigger step sizes may cause inaccuracy. The sparsity range depends on the applications. Generally, it can be set as $[1, p/3]$, where $p$ is the dimension of the sparse data. Algorithm 3 describes the proposed AdaDGS recovery algorithm.

For practical sparse recovery, choosing a halting condition is always a troublesome problem. We find that there is no optimal way to choose one halting condition for all cases. There are two preferred choices. The first one is just as that in CoSaMP [97]. During the iterations, the proposed DGS is stopped when the residual norm in the current iteration is not smaller than that in the last iteration. In practice, we find that it makes the recovery algorithm very fast. The recovered result is good enough for many practical sparse recovery problems although it may be not optimal. Thus, it is used in the inner loop. Another halting condition is suggested by the GPSR algorithm [39]: the program is stopped when the relative change of the norm of the recovered data between two consecutive iterations is smaller than a certain threshold. The underlying intuition is that it is not worth taking more iterations if the resulting improvement is too small. The outer loop in AdaDGS uses this stopping scheme.
4.4 DGS for Compressive Sensing

For quantitative evaluation, the recovery error is defined to indicate the difference between the estimation $x_{est}$ and the ground-truth $x$: $\|x_{est} - x\|_2 / \|x\|_2$. All experiments are conducted on a 3.2GHz PC in Matlab environment.

4.4.1 1D Simulated Signals

In the first experiment, we randomly generate a 1D $G(k, q)$-sparse signal with values $\pm 1$, where $p = 512$, $k = 64$ and $q = 4$. The projection matrix $\Phi$ is generated by creating a $n \times p$ matrix with i.i.d. drawn of a Gaussian distribution $N(0; 1)$, and then the rows of $\Phi$ are normalized to the unit magnitude. Zero-mean Gaussian noise with standard deviation $\sigma = 0.01$ is added to the measurements. Figure 4.3 shows one generated signal and its recovered results by different algorithms when $n = 3k = 192$. As the measurement number $n$ is only 3 times of the sparsity number $k$, both of other algorithms can not obtain good recovery results, whereas the DGS obtains almost perfect recovery results with the least running time. To study how the measurement number $n$ affects the recovery performance, we change the measurement number and record the recovery results by different algorithms. To reduce the randomness, we execute the experiment 100 times for each of the measurement numbers in testing each algorithm. Figure 4.4 shows the performance of 5 algorithms with increasing measurements in terms of the recovery error and running time. Overall, the DGS obtains the best recovery performance with the least computation; the recovery performance of GPSR, SPGL1-Lasso and SP is close; and the $l^1$-norm minimization based GPSR and SPGL1-Lasso require more computation than greedy algorithms such as the OMP, SP and our DGS. In the three greedy algorithms, the OMP has the worst recovery performance. All these experimental results are consistent with our theorem: the proposed DGS algorithm can achieve better recovery performance for DGS data with far fewer measurements and less computation complexity.

4.4.2 2D Color Images

To validate the proposed recovery algorithm on 2D images, we randomly generate a 2D $G(k, q)$-sparse color image by putting four digits in random locations, where $p = H \times W = 48 \times 48$, $k = 152$ and $q = 4$. The projection matrix $\Phi$ and noises are generated with the similar method as that for 1D signal. The $G(k, q)$-sparse color image has a special property: the R, G and B channels share a common support set, while the nonzero coefficients have dynamic group clustering trends within each channel. Thus, the recovery of DGS color images can be considered as a
Figure 4.3: Recovery results of 1D data. (a) Original data; (b) GPSR (error is 0.5173 and time is 0.1847 seconds); (c) SPGL1-Lasso (error is 0.4021 and time is 1.1497 seconds); (d) OMP (error is 1.0270 and time is 0.1422 seconds); (e) SP (error is 0.6143 and time is 0.1100 seconds); (f) DGS recovery (error is 0.0178 and time is 0.0605 seconds).

Figure 4.4: Recovery errors vs. measurement numbers: a) recovery errors; (b) running times
3-task CS recovery problem. As for the input parameters of DGS in this case, we just need to set $J_y$ as 3 and keep other default parameters unchanged. Considering the MCS is specially designed for multi-task CS problems, we will compare it with DGS and SP. Figure 4.5 shows one example 2D $G(k,q)$-sparse color image and the recovered results by different algorithms when $n = 440$. Figure 4.6 shows the performance of the three algorithm, averaged over 100 random runs for each sample size. The DGS achieves the best recovery performance with far less computation. It is easily understood because DGS exploits three priors for recovery: (1) the three color channels share a common support set, (2) there are dynamic group clustering trends within each color channel and (3) sparsity prior exists in each channel; thus it achieves better results than MCS, which only uses two priors. The SP is the worst since it only uses one prior. This experiment clearly demonstrates: the more valid priors are used for sparse recovery, the more accurate results we can achieve. That is the main reason why DGS, MCS and SP obtained the best, good and the worst recovery results. Figure 4.6 (b) shows the comparison of running times by the three algorithms. It is not surprising that the running times with DGS are always far less than those with MCS and a little less than those with SP for all measurement numbers.

![Figure 4.5: Recovery results of a 2D color image: (a) original color image; (b) recovered image with MCS [72] (error is 0.8399 and time is 29.2656 seconds); (c) recovered image with SP [29] (error is 0.7605 and time is 1.6579 seconds) and (d) recovered image with DGS (error is 0.1176 and time is 1.0659 seconds).](image)

4.5 DGS for Background Subtraction

4.5.1 Background Subtraction

Background subtraction is an important pre-processing step in video monitoring applications. There exist a lot of methods for this problem. The Mixture of Gaussians (MoG) background model assumes the color evolution of each pixel can be modelled as a MoG and are widely
Figure 4.6: Recovery performance: a) errors; (b) running times.
used on realistic scenes [120]. Elgammal et al. [37] proposed a non-parametric model for the background under similar computational constraints as the MoG. Spatial constraints are also incorporated into their model. Sheikh and Shah consider both temporal and spatial constraints in a Bayesian framework [116], which results in good foreground segmentations even when the background is dynamic. The model in [93] also uses a similar scheme. All these methods only implicitly model the background dynamics. In order to better handle dynamic scenes, some recent work [94, 156] explicitly model the background as dynamic textures. Most dynamic texture modeling methods are based on the Auto Regressive and Moving Average (ARMA) model, whose dynamics is driven by a linear dynamic system (LDS). While this linear model can handle background dynamics with certain stationarity, it will cause over-fitting for more complex scenes.

4.5.2 Algorithm

The inspiration for our AdaDGS background subtraction came from the success in online DT video registration based on the sparse representation constancy assumption (SRCA) [59]. The SRCA states that a new coming video frame should be represented as a linear combination of as few preceding image frames as possible. As a matter of fact, the traditional brightness constancy assumption seeks that the current video frame can be best represented by a single preceding frame, while the SRCA seeks that the current frame can be best sparsely represented by all preceding image frames. Thus, the former can be thought as a special case of SRCA.

Suppose a video sequence consists of frames $I_1, \ldots, I_T \in \mathbb{R}^m$. Without loss of generality, we can assume that background subtraction has already been performed on the first $t$ frames. Let $A = [I_1, \ldots, I_t] \in \mathbb{R}^{m \times t}$. Denote the background image and the background subtracted image by $b$ and $f$, respectively, for $I_{t+1}$. From the introduction in Section 4.3.1, we know that $f$ is dynamic group sparse data with unknown sparsity number $k_f$ and group structure. According to SRCA, we have $b = Ax$, where $x \in \mathbb{R}^t$ should be $k_x$-sparse vector and $k_x << t$.

Let $\Phi = [A, I] \in \mathbb{R}^{m \times (t+m)}$, where $I \in \mathbb{R}^{m \times m}$ is an identity matrix. Then, we have:

$$I_{t+1} = Ax + f = [A, I] \begin{bmatrix} x \\ f \end{bmatrix} = \Phi z$$

(4.9)

where $z \in \mathbb{R}^{t+m}$ is the DGS data with unknown sparsity $k_x + k_f$.

Background subtraction is thus formulated as the following AdaDGS recovery problem:

$$(x_0, f_0) = \text{argmin} \|z\|_0, \quad \|I_{t+1} - \Phi z\|^2 < \varepsilon$$

(4.10)
which can be efficiently solved by the proposed AdaDGS recovery algorithm. Similar ideas are
used for face recognition robust to occlusion [141]. It is worth mentioning that the coefficients in
weight \( w \) corresponding to the \( x \)-related part are randomly sparse while those corresponding to \( f \) are dynamic
group sparse. During the DGS approximation pruning, we thus can set those coefficients in
weight \( w \) for the \( x \)-related part as zeros and those for \( f \) as nonzeros. Since we do not know
the sparsity number \( k_x \) and \( k_f \), we can set sparsity ranges for them respectively and run the
AdaDGS recovery algorithm until the halting condition is true. Then, we can obtain the
optimized background subtracted image \( f \) and background image \( b = Ax \). For long video
sequences, it is impractical to build a model matrix \( A = [I_1, \ldots, I_t] \in \mathbb{R}^{m \times t} \), where \( t \) denotes the
last frame number. In order to cope with this case, we can set a time window width parameter
\( \tau \). We then build the model matrix, \( A = [I_{t-\tau+1}, \ldots, I_t] \in \mathbb{R}^{m \times (t-\tau)} \), for the \((t+1)\) frame, which
can avoid the memory requirement blast for a long video sequence. The complete algorithm for
AdaDGS based background subtraction is summarized in Algorithm 4.

### Algorithm 4 AdaDGS Background Subtraction

1: **Input:** The video sequence \( I_1, \ldots, I_T \), the number \( t \) which means \( 1^{st} \sim t^{th} \) have been
   performed background subtraction, the time window width \( \tau \leq t \)
2: **for** all \( j = t+1, \ldots, n \) **do**
3:   Set \( A = [I_{j-\tau}, \ldots, I_{j-1}] \) and form \( \Phi = [A, I] \)
4:   Set \( y = I_j \) and the sparsity ranges/step-sizes
5:   \((x_0, f_0) = AdaDGS(\Phi, y)\)
6: **end for**
7: **Output:** Background subtracted images

### 4.5.3 Experiments

Background subtracted images are typical \( G(k, q) \)-sparse data. They generally correspond to
the interested foreground objects. Compared with the whole scene, they tend to be not only
spatially sparse but also cluster into dynamic groups, although the sparsity number and group

![Figure 4.7: Background subtraction in video surveillance where foreground is detected as anomaly. The columns of \( A \) represent the previous background only images](image)
structures are not known. As we know, the sparsity number must be provided in most of current recovery algorithms, which make them impractical for this problem. In contrast, the proposed AdaDGS can apply well to this task since it not only can automatically learn the sparsity number and group structures but also is a fast enough greedy algorithm.

The first experiment is designed to validate the advantage of the AdaDGS model. We test the proposed algorithm on Zhong’s dataset [156]. The background subtracted images can be directly obtained with the proposed AdaDGS. The corresponding binary mask of these images are obtained with the simple threshold. The Zhong’s results with robust Kalman model are also shown for comparisons. Figure 4.8 shows the results. Note that all results with AdaDGS are not post-processed with morphological operations and the results are directly the solutions of the optimization problem in Equation 4.10. It is clear that our AdaDGS produces clean background subtracted images, which shows the advantages of the DGS model. Figure 4.9 and Figure 4.11 show the background subtraction results on two other videos [37, 94]. Note that our results without postprocessing can compete with others with postprocessing. The results show the proposed AdaDGS model can handle well highly dynamic scenes by exploiting the effective sparsity optimization scheme.

![Figure 4.8: Results on the Zhong’s dataset (a) original frame; (b) background subtracted image with AdaDGS; (c) the binary mask with AdaDGS and d) with robust Kalman model [156].](image-url)
Figure 4.9: Results on the Elgammal’s dataset. (a) original frame; (b) with AdaDGS; (c) with KDE model [37] (d) with MoG [120].

Figure 4.10: Results on Monnet’s dataset. (a) original frame; (b) with AdaDGS; (c) with Monnet’s method [94] and (d) with MoG [120].

Figure 4.11: Results on a helicopter video. (a) original frame; (b) with AdaDGS; (c) with MoG [120] and (d) with robust Kalman model [156].
4.6 DGS for Visual Tracking

4.6.1 Tracking

Tracking is to estimate the state of the moving target in the coming observed sequences. This topic is interesting for many industrial applications, such as surveillance, traffic monitoring, vehicle navigation, video indexing, etc. Accurate tracking of general object in a dynamic environment is still a challenging problem. [144, 146].

Online adaptive tracking method is intensively investigated in recent literature. Grabner et al [48] propose to update the feature selection incrementally using the training samples gathered from current tracking result, which may lead to potential target drifting because of accumulated errors. Semi-online boosting [49] was proposed to incrementally update the classifier using unlabeled and labeled data together to avoid the target drifting. Multiple Instance Learning boosting method (MILBoosting) [5] put all samples into bags and labeled them with bag labels. The positive bag is required to contain at least one real positive, while the negative bags have only negative samples. The drifting problem is handled in their method since the true target included in positive bag is learned implicitly. The target is represented as a single online learned appearance model in incremental visual tracking (IVT) [107]. As single appearance model is argued to be not sufficient to present the target in a dynamic environment, multiple appearance models are also proposed to be incrementally learned during the tracking in [147]. Online updating is proven to be an important step in adaptive tracking and is also used in our algorithm. Sparse representation was introduced for tracking in [92]. The target candidate is represented as a linear combination of the learned template set composed of both target templates and trivial templates which have only one nonzero element. The assumption is that good target candidate can be sparsely represented by both the target templates and the trivial templates. This sparse optimization problem is solved as a $l_1$ minimization problem with nonnegative constraints.

In practical applications, we observed that the target can usually be represented by templates sparsely and only part of the features, which can discriminate the target and background, are necessary to identify the target. Motivated by [92], considering existing problems and our observations, we proposed a robust and fast tracking algorithm with two stage sparse optimization. The algorithm starts from feature selection by solving a dynamic group sparsity (DGS) [60] optimization problem. The DGS is then performed on the selected feature space for sparse reconstruction of the target. These two sparsity problems are optimized jointly and the final results are obtained by Bayesian inference.
**Algorithm 1**: Tracking with two stage sparsity optimization

**Input**: Target’s initial state $\chi_0$, sparsity parameter $K_0$ for feature selection, $K_1$ and $K_2$ for target template and trivial template.

**Initialize**: Construct $n$ training samples $\{X \in \mathbb{R}^{n \times p}, L \in \mathbb{R}^{n \times 1}\}$, where $X$ is the sample matrix, $L$ is the label and $p$ is the dimension of the feature vector.

1. For each frame $t = 1 : T$ in the video where $T$ is the total number of frames:
   2. Perform DGS to solve $w^* = \text{argmin}_w ||Xw - L||_2$, subject to: $|w|_0 \leq K_0$ (when $t = 1$ we will use the initializations).
   3. Construct diagonal matrix $W$, $W_{i,i} = \begin{cases} 1, & w^*_i \neq 0 \\ 0, & \text{otherwise}; \end{cases}$
   4. Generate $N$ candidate samples $y_i$ in state $\chi^*_t$.
   5. For each $y_i, i = 1 : N$
     6. Let $W' \in \mathbb{R}^{K_0 \times p}$ as the matrix contains all non-zero rows of $W$,
     7. $\Phi' = W'\Phi, y'_i = W'y_i, \text{ and } f' = W'f$,
     8. perform DGS to solve
     9. $(\alpha^*, f^*) = \text{argmin}_{\alpha, f} ||\Phi' W' \begin{bmatrix} \alpha \\ f \end{bmatrix} - y'_i||_2$, subject to:
        $||\alpha||_0 \leq K_1$
        $||f||_0 \leq K_2$.
     10. $\epsilon_i = ||\Phi' \alpha^* - y'_i||_2$.
     11. $p(z_t|\chi^*_t) = \exp(-\epsilon_i)$.
     12. end for
     13. $\chi^*_t = \text{argmax}_{\chi^*_t} p(\chi^*_t|z_{1:t})$.
     14. Update the training set and template library with tracking results.
     15. end for

4.6.2 Algorithm

Given the learned target template library $\Phi \in \mathbb{R}^{p \times m}$, where $m$ is the number of templates and $p$ is the dimension of the features. Let $\Phi_1 = [\Phi, f]$ and $\alpha_1 = \begin{bmatrix} \alpha \\ f \end{bmatrix}$ where $\alpha$ represents the sparse coefficient vector and $f$ denotes the occlusion, the candidate sample $y$ is sparsely reconstructed from $\Phi$ by minimizing the $l_2$ errors and finding $\alpha$ with $K_1$ nonzero components and $f$ with $K_2$ nonzero components using greedy method:

$$\alpha_1 = \text{argmin}_{\alpha, f} ||\Phi_1 \alpha_1 - y||_2, \text{ while } ||\alpha||_0 \leq K_1 \text{ and } ||f||_0 \leq K_2.$$  \hfill (4.11)

Equation (4.11) can be solved efficiently when the dimension of the feature space and candidate searching space are small. However, it is computationally expensive for very high dimensional data, which make it unsuitable if advanced image features are used. Because only some of the features, which can discriminate the target and background, are necessary to identify the target, we argued that the effective dimension of the feature space can be decreased to $K_0$ dimension with diagonal matrix $W$. The number of nonzero components in $W$ is not larger than $K_0$. The $i$-th feature is activated if $W_{ii}$ is nonzero. Given $n$ available samples $X \in \mathbb{R}^{n \times p}$ and their labels $L \in \mathbb{R}^{n \times 1}$, The joint sparse solution can be found:
\[(\alpha_1, W) = \arg\min_{\alpha_1, W} \lambda ||W\Phi_1\alpha_1 - Wy||_2 \]
\[+ \beta F(W, X, L) + \tau_1 ||\alpha_1||_1 + \tau_2 ||\text{diag}(W)||_1 \]
\[(4.12)\]

where \(F(W, X, L)\) is the loss function in the selected feature space for training dataset and samples in current frame. The \(\tau_1\) and \(\tau_2\) are the sparse parameters. As we explained before, the parameters \(\tau_1\) and \(\tau_2\) in (4.12) have no physical meaning and therefore it is difficult to tune their values. In our algorithm, we apply greedy algorithm to directly solve the original \(l_0\) minimization problem for sparse representation. In this way (4.12) can be rewritten as:

\[(\alpha_1, W) = \arg\min_{\alpha_1, W} \lambda ||W\Phi_1\alpha_1 - Wy||_2 + \beta F(W, X, L), \]
\[\text{subject to: } ||\text{diag}(W)||_0 \leq K_0, ||\alpha||_0 \leq K_1 \text{ and } ||f||_0 \leq K_2. \]
\[(4.13)\]

As it is hard to find an optimum solution for (4.12) when both \(\alpha_1\) and \(W\) are unknown, we solve (4.13) using two stage dynamic group sparsity optimization with greedy method. The first stage is to select the sparse set of features that are most discriminative in separating the target from background. Then the generative likelihood of each sample is estimated in the second stage with sparse representation. The details of the algorithm are shown in Algorithm 1. We will explain each stage in the following sections.

**Feature selection**

Given a set of training data \(X = \{x_i \in \mathbb{R}^{1 \times p}\} \) with \(L = \{l_i\}, i = 1 \ldots n\) as the labels. The term \(F(W, X, L)\) in equation 4.12 is defined as

\[F(W, X, L) = e^{-\sum_{i=1}^{n}(x_iw)l_i}, \]
\[(4.14)\]

where \(w \in \mathbb{R}^{p \times 1}\) is a sparse vector. The \(j\)-th feature is selected if \(w_j \neq 0\). The solution to minimize \(F(W, X, L)\) can be found by solving the following sparse problem

\[w^* = \arg\min_w ||Xw - L||, \text{subject to: } ||w||_0 \leq K_0 \]
\[(4.15)\]

where \(K_0\) is the max number of features to be selected. Here we want to emphasize that using greedy method for optimization, the parameter \(K_0\) has a specific physical meaning - it is the number of features we plan to select. Considering Haar-like features, we do have the spatial relationship between neighborhood features. For example, if a small region is occluded, the features extracted from this region will tend to be treated as a group in sparse optimization.
Let $N_w(i, j)$ as the value of $j$-th neighbor of $i$-th feature, the support set is pruned based on $Z$

$$z_i = w_i^2 + \sum_{j=1}^{s} \theta_j^2 N_w^2(i, j), \ i = 1 \ldots p \quad (4.16)$$

in DGS taking the neighborhood relationship into consideration, where $\theta$ is the weight of neighbors. With the optimal $w$ found by DGS, The diagonal matrix $W$ can be constructed as

$$W_{j,j} = \begin{cases} 1, w_j \neq 0 \\ 0, otherwise; \end{cases} \quad (4.17)$$

Benefiting from the sparse solution to (4.15), we will be able to use advanced high dimensional features without sacrificing the efficiency of the algorithm. The other benefit is the object selection in the target region. The target templates usually contain some background features which are not linear. By doing discriminative feature selection, features from background pixels in the target templates are eliminated. The target template library is therefore more efficient and robust.

**Sparse Reconstruction**

After we calculate the weighting matrix $W$, the $\alpha$ and $f$ in equation (4.12) can be found in the second stage

$$(\alpha, f) = \arg\min_{\alpha, f} ||W \Phi_1 \alpha_1 - W y||, \text{ subject to: } ||\alpha||_0 \leq K_1 \text{ and } ||f||_0 \leq K_2. \quad (4.18)$$
Figure 4.13: The second stage: feature selection to accelerate the tracking process
where $\Phi_1 = [\Phi, I]$ and $\alpha_1 = \begin{bmatrix} \alpha \\ f \end{bmatrix}$. Let $W' \in \mathbb{R}^{K_0 \times p}$ as the matrix contains all nonzero rows of $W$. We define $\Phi' = W'\Phi$ and $y' = W'y$. Please notify that in this step we already reduced the feature dimension from $p \times m$ to $K_0 \times m$ where $m$ is the number of templates in the target library. In this stage the following equation is solved

$$(\alpha^*, f^*) = \arg\min_{\alpha, f} \left| \left| \Phi', W' \right| \left| \begin{bmatrix} \alpha \\ f \end{bmatrix} - y' \right|, \right. \quad \text{subject to: } \begin{align*} ||\alpha||_0 &\leq K_1 \\ ||f||_0 &\leq K_2 \end{align*} \quad (4.19)$$

Here the sparsity parameter $K_1$ and $K_2$ have clear physical meaning, where $K_1$ controls the sparsity of the target template representation and $K_2$ controls the tolerance of occlusion. Then the likelihood of the testing sample $y$ as target is $e^{-|\Phi'\alpha^* - y'|^2}$ and the final result is obtained by maximizing the probability.

Once the tracking result is confirmed, the template library is incrementally updated as [92]. The samples with high likelihood and near the target are added to the training set as positive while the others are added as negative samples. This procedure is repeated for each frame in a whole sequence. The joint optimization of the two stage sparsity problem thus provides a fast, robust and accurate tracking result.

### 4.6.3 Experiments

We conduct experiments on five challenging sequences with 3217 frames in total. The method is compared with three state-of-art tracking methods named L1 tracker (L1) [92], Incremental Visual Tracking (IVT) [107] and Multiple Instance Learning (MIL) [5]. The tracking results of the compared algorithms are obtained by running the binaries or source code provided by their authors using the same initial positions. The source code of L1, IVT, MIL can be obtained from the URLs \(^2\) \(^3\) \(^4\). The first, second, third and fourth sequences were obtained from [107], and the fifth sequence was downloaded from [5].

The first sequence was captured in an open road environment. The tracking results of the 4, 56, 200, 238, 309 are presented in Figure 4.14. The L1 starts to show some drifting on the 56-th frame. The MIL starts to show some target drifting (on the 200-th frame) and finally loses the target (the 238-th frame). IVT can track this sequence quite well. The target was successfully tracked using our proposed algorithm during the entire sequence.

\(^2\)http://www.ist.temple.edu/~hbling/code_data.htm
\(^3\)http://www.cs.toronto.edu/~dross/ivt/
\(^4\)http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml
The second sequence is to track a moving face. Five (2, 47, 116, 173, and 222) frames are presented in Figure 4.15. The L1 algorithm fails to track the target when there are both pose and scale changes, as shown in the 116-th frame. The MIL method can roughly capture the position of the object, but does have some target drift problems, especially in the 173-th and 222-th frame. Our proposed two stage sparse tracking algorithm can track the moving face accurately through the whole sequence while the IVT produces some errors, especially on the 222-th frame.

The third image sequence with frame 2, 241, 459, 611, and 693 is shown in Figure 4.18. The L1 method starts to have some drifting problem from roughly the 200-th frames, shown in the 241-th and 459-th frame. The MIL algorithm provides very good tracking results in this sequence. IVT fails to follow the object on the 611-th frame after major pose variation and can not be recovered. Our algorithm provides robust and accurate tracking result for the whole sequence.

In the fourth sequence, the vehicle was driven in a very dark environment and captured from another moving vehicle. The 2, 35, 208, 239, 378 frames are presented in Figure 4.17. The L1 algorithm starts to fail to track the target from the 35-th frame. The MIL can roughly capture the position of the object at the beginning, but the bounding box starts to drift from the 208-th frame, due to illumination change. IVT can track the target through the whole video sequence.
Figure 4.15: The tracking results of a moving face sequence, which has large pose variation, scaling, and illumination changes. The order of the row sequences is the same as Figure 4.14. but it is not as accurate as our results, which can be found in the 378-th frame.

The results of the Face sequence are shown in Figure 4.18. In this sequence we show the robustness of our algorithm in handling occlusion. The frame indexes are 10, 427, 641, 713, and 792. Starting from the 641-th frame, our method performs consistently better than other methods.

4.7 Summary

This chapter introduces a special model of structured sparsity called dynamic group sparsity. The model assumes that the nonzero coefficients are often not random but tend to be clustered. Intuitively, we can obtain better results in these cases by reasonably utilizing both clustering
Motivated by this idea, we first developed a new greedy sparse recovery algorithm (DGS), which prunes data residues in the iterative process according to both sparsity and group clustering priors rather than only sparsity as in previous methods. We also developed the AdaDGS to adaptively learn the dynamic group structure and the sparsity number if they are not available in the practical applications. It has been used for compressive sensing and can recover stably sparse data with clustering trends using far fewer measurements and computations than current state-of-the-art algorithms with provable guarantees. It was successfully used to obtain robust results for video background subtraction in dynamic scene. We also developed a robust and fast method for visual tracking based on the DGS. All these successful applications confirmed the advantages of the structured sparsity.
Figure 4.17: The tracking results of the car sequence in a dark environment. This sequence has low resolution and poor contrast, which introduce some landmark ambiguity. Results of our algorithm, MIL, L1, and IVT are shown from top to bottom rows respectively.

Figure 4.18: The tracking results of a face sequence, which includes a lot of pose variations, partial or full occlusions. Results of our algorithm, MIL, L1, and IVT are shown from top to bottom rows respectively.
Chapter 5
Applications in Magnetic Resonance Imaging

In this Chapter, we propose an efficient algorithm for Magnetic Resonance (MR) image reconstruction. The algorithm minimizes a linear combination of three terms corresponding to a least square data fitting, total variation (TV) and $L_1$ norm regularization. This has been shown to be very powerful for the MR image reconstruction. First, we decompose the original problem into $L_1$ and TV norm regularization subproblems respectively. Then, these two subproblems are efficiently solved by existing techniques. Finally, the reconstructed image is obtained from the weighted average of solutions from two subproblems in an iterative framework. We compare the proposed algorithm with previous methods in term of the reconstruction accuracy and computation complexity. Numerous experiments demonstrate the superior performance of the proposed algorithm for compressed MR image reconstruction. This work was presented under a slightly modified form in [64].

5.1 Introduction

Magnetic Resonance Imaging has been widely used in medical diagnosis because of its non-invasive manner and excellent depiction of soft tissue changes. It has been accepted by a lot of radiologists and oncologists.

Figure 5.1: The MRI machines.
An important research topic in MR imaging is how to reduce the scanning time for acquiring MR images with high quality. As we know, the longer scanning duration will inevitably cause more physiological motion artifacts and increase the patient discomfort. On the other hand, there exists a fact in MR imaging that the scanning time linearly depends on the number of the sampled K-space data. Therefore, reducing the scanning time means that acquiring fewer samples from K-space data. According to the Nyquist criterion, this means reducing scanning time will inevitably sacrifice the image quality. Recent developments in compressive sensing [23][32] make it possible to reduce the scanning time without sacrifice of image quality. It is based on the sparsity nature of the practical signals in a transform domain and enable it to reconstruct signals from very few samples.[23], Candes et al. applied the compressive sensing to image reconstruction on the Shepp-Logan phantom from K-space samples. Perfect reconstruction results are obtained from 9% samples of the K-space, which shows the potential power of the compressive sensing for MR image reconstruction.

Suppose $x$ is a MR image and $R$ is a partial Fourier transform, the sampling measurement $b$ of $x$ in K-space is defined as $b = Rx$. The compressed MR image reconstruction problem is to reconstruct $x$ given the measurement $b$ and the sampling matrix $R$. Motivated by the compressive sensing theory, Lustig et al. [83] proposed their pioneering work for the MR image reconstruction. Their methods can reconstruct both phantoms and practical anatomical images very well with only 20% sampling. Since the real MR images are less sparse than phantom images, it is necessary to use larger sampling rate to obtain good reconstruction [23]. Their method can effectively reconstruct MR images with only 20% sampling. The improved results were obtained by having both a wavelet transform and a discrete gradient in the objective, which is formulated as follows:

$$\hat{x} = \arg \min_x \left\{ \frac{1}{2} \| Rx - b \|^2 + \alpha \| x \|_{TV} + \beta \| \Phi x \|_1 \right\} \quad (5.1)$$

where $\alpha$ and $\beta$ are two positive parameters, $b$ is the undersampled measurements of K-space data, $R$ is a partial Fourier transform and $\Phi$ is a wavelet transform. It is based on the fact that the piecewise smooth MR images of organs can be sparsely represented by the wavelet basis and should have small total variations. The TV was defined discretely as $\| x \|_{TV} = \sum_i \sum_j (\| \nabla_1 x_{ij} \|^2 + \| \nabla_2 x_{ij} \|^2)$ where $\nabla_1$ and $\nabla_2$ denote the forward finite difference operators on the first and second coordinates, respectively. Since both $L_1$ and TV norm regularization terms are nonsmooth, this problem is very difficult to solve. The conjugate gradient (CG) [83] and PDE [56] methods were used to attack it. However, they are very slow and impractical for real MR images. Computation became the bottleneck that prevented this good model (5.1) from
being used in practical MR image reconstruction. Therefore, the key problem in compressed MR image reconstruction is thus to develop efficient algorithms to solve problem (5.1) with nearly optimal reconstruction accuracy.

Other methods tried to reconstruct compressed MR images by performing $L_p$-quasinorm ($p < 1$) regularization optimization [145, 24, 25]. Although they may achieve a little bit of higher compression ratio, these nonconvex methods do not always give global minima and are also relatively slow. Trzasko et al. [130] used the homotopic nonconvex $L_0$-minimization to reconstruct MR images. They created a gradual nonconvex objective function which may allow global convergence with designed parameters. It was faster than those $L_p$-quasinorm regularization methods. However, it still needed $1 - 3$ minutes to obtain reconstructions of $256 \times 256$ images in MATLAB on a 3 GHz desktop computer. Recently, two fast methods were proposed to directly solve (5.1). In [85], Ma et al. proposed an operator-splitting algorithm (TVCMRI) to solve the MR image reconstruction problem. In [143], a variable splitting method (RecPF) was proposed to solve the MR image reconstruction problem. Both of them can replace iterative linear solvers with Fourier domain computations, which can gain substantial time savings. In MATLAB on a 3 GHz desktop computer, they can be used to obtain good reconstructions of $256 \times 256$ images in ten seconds or less. They are two of the fastest MR image reconstruction methods so far.

Model (5.1) can be interpreted as a special case of general optimization problems consisting of a loss function and convex functions as priors. Two classes of algorithms to solve this generalized problem are operator-splitting algorithms and variable-splitting algorithms.

The operator-splitting algorithm is to search an $x$ to make the sum of the corresponding
maximal-monotone operators equal to zero. Forward-Backward schemes are widely used in operator-splitting algorithms [41, 28, 132]. The Iterative Shrinkage-Thresholding Algorithm (ISTA) and Fast ISTA (FISTA) [12] are two important Forward-Backward methods. They have been successfully used in signal processing [12, 11], matrix completion [125] and multi-task learning [74]. To handle the case of more than one convex function, Spingarn [118] reduces the sum of multiple maximal monotone operators to the sum of two maximal monotone operators by defining new subspaces and operators, and then applies a Douglas-Rachford splitting algorithm to solve the new problem. The general projective splitting methods are proposed to search a point in the extended solution set [36].

The variable splitting algorithm is another choice to solve such problems based on the combination of alternating direction methods (ADM) under an augmented Lagrangian framework: 1) splitting the variable $x$ into $m + 1$ variables by introducing $m$ new variables (assuming $m$ is the number of convex functions), where each new variable corresponds to one convex function; 2) applying the augmented Lagrangian method to this problem for each variable; 3) minimizing the decomposed augmented-Lagrangian function using the ADMs to iteratively obtain the solutions. It is firstly used to solve the numerical PDE problem in [42, 44]. Tseng and He et al. extend it to solve variational inequality problems [131, 53]. Wang et al. [137] incorporate the equality constraints into the objective function and apply an ADM to the new penalized function for obtaining better image reconstruction, which shows that the ADMs are very efficient for solving TV regularization problems. They are recently proposed to solve semi-definite programming (SDP) problems and outperform previous interior-point methods on some structured SDP problems [139, 87]. They are also applied to solve the sparse covariance selection problem [149] and matrix decomposition problem [150]. The Multiple Splitting Algorithm (MSA) and Fast MSA (FaMSA) have been recently proposed to efficiently solve (5.1), while all convex functions are assumed to be smooth [45].

However, all these above-mentioned algorithms can not efficiently solve (5.1) with provable convergence complexity. Moreover, none of them can provide the complexity bounds of iterations for their problems, except ISTA/FISTA in [12] and MSA/FaMSA in [45]. Both ISTA and MSA are first order methods. Their complexity bounds are $O(1/\epsilon)$ for $\epsilon$-optimal solutions. Their fast versions, FISTA and FaMSA, have complexity bounds $O(1/\sqrt{\epsilon})$, which are inspired by the seminal results of Nesterov and are optimal according to the conclusions of Nesterov [99, 100]. However, both ISTA and FISTA are designed for simpler regularization problems and can not be applied efficiently to the composite regularization problem (5.1) using both $L_1$ and TV norm. While the MSA/FaMSA assume that all convex functions are smooth. It makes
them unable to directly solve the problem (5.1) as we have to smooth the nonsmooth function first before applying them. Since the smooth parameters are related to $\epsilon$, the FaMSA with complexity bound $O(1/\sqrt{\epsilon})$ requires $O(1/\epsilon)$ iterations to compute an $\epsilon$-optimal solution, which means that it is not optimal for this problem.

In this section, we propose a new optimization algorithm for MR image reconstruction. It is based on the combination of both variable and operator splitting techniques. We decompose the hard composite regularization problem (5.1) into two simpler regularization subproblems by: 1) splitting variable $x$ into two variables $\{x_i\}_{i=1,2}$; 2) performing operator splitting to minimize total variation regularization and $L_1$ norm regularization subproblems over $\{x_i\}_{i=1,2}$ respectively and 3) obtaining the solution $x$ by linear combination of $\{x_i\}_{i=1,2}$. This includes both variable splitting and operator splitting. We call it the Composite Splitting Algorithm (CSA). Motivated by the effective acceleration scheme in FISTA [12], the proposed CSA is further accelerated with an additional acceleration step. Numerous experiments have been conducted on real MR images to compare the proposed algorithm with previous methods. Experimental results show that it impressively outperforms previous methods for the MR image reconstruction in terms of both reconstruction accuracy and computation complexity.

Some contents introduced in this section has been published in MICCAI’10 [62].

5.2 Methodology

5.2.1 Related Acceleration Algorithm

In this section, we briefly review the FISTA in [12], since our methods are motivated by it. FISTA considers to minimize the following problem:

$$\min \{F(x) \equiv f(x) + g(x), x \in \mathbb{R}^p\} \quad (5.2)$$

where $f$ is a smooth convex function with Lipschitz constant $L_f$, and $g$ is a convex function which may be nonsmooth.

**$\epsilon$-optimal Solution:** Suppose $x^*$ is an optimal solution to (5.2), $x \in \mathbb{R}^p$ is called an $\epsilon$-optimal solution to (5.2) if $F(x) - F(x^*) \leq \epsilon$ holds.

**Gradient:** $\nabla f(x)$ denotes the gradient of the function $f$ at the point $x$.

**The proximal map:** given a continuous convex function $g(x)$ and any scalar $\rho > 0$, the proximal map associated with function $g$ is defined as follows [12][11]:

$$\text{prox}_\rho(g)(x) := \arg \min_u \{g(u) + \frac{1}{2\rho} \|u - x\|^2\} \quad (5.3)$$
Algorithm 5 FISTA [12]

Input: $\rho = 1/L_f$, $r^1 = x^0$, $t^1 = 1$

for $k = 1$ to $K$

do

$x_g = r^k - \rho \nabla f(r^k)$

$x^k = \text{prox}_\rho(g)(x_g)$

$k^{k+1} = \frac{1+\sqrt{1+4\rho^2}}{2}$

$r^{k+1} = x^k + \frac{r^k}{k+1}(x^k - x^{k-1})$

end for

Theorem 5.2.1 (Theorem 4.1 in [12]: Suppose $\{x^k\}$ and $\{r^k\}$ are iteratively obtained by the FISTA, then, we have

$$F(x^k) - F(x^*) \leq \frac{2L_f||x^0 - x^*||^2}{(k+1)^2}, \forall x^* \in X_*$$

The efficiency of the FISTA highly depends on being able to quickly solve its second step $x^k = \text{prox}_\rho(g)(x_g)$. For simpler regularization problems, it is possible, i.e, the FISTA can rapidly solve the $L_1$ regularization problem with cost $O(p \log(p))$ [12] (where $p$ is the dimension of $x$), since the second step $x^k = \text{prox}_\rho(\beta \|x\|_1)(x_g)$ has a closed form solution; It can also quickly solve the TV regularization problem, since the step $x^k = \text{prox}_\rho(\alpha \|x\|_{TV})(x_g)$ can be computed with cost $O(p)$ [11]. However, the FISTA cannot efficiently solve the composite $L_1$ and TV regularization problem (5.1), since no efficient algorithm exists to solve the step $x^k = \text{prox}_\rho(\alpha \|x\|_{TV} + \beta \|\Phi x\|_1)(x_g)$.

To solve the problem (5.1), the key problem is thus to develop an efficient algorithm to solve problem (5.4). In the following section, we will show that a scheme based on composite splitting techniques can be used to do this.

5.2.2 CSA and FCSA

From the above introduction, we know that, if we can develop a fast algorithm to solve problem (5.4), the MR image reconstruction problem can then be efficiently solved by the FISTA, which obtains an $\epsilon$-optimal solution in $O(1/\sqrt{\epsilon})$ iterations. Actually, problem (5.4) can be considered as a denoising problem:

$$x^k = \arg\min_x \left\{ \frac{1}{2}||x - x_g||^2 + \rho\alpha \|x\|_{TV} + \rho\beta \|\Phi x\|_1 \right\}.$$  \hspace{1cm} (5.5)

We use composite splitting techniques to solve this problem: 1) splitting variable $x$ into two variables $\{x_i\}_{i=1,2}$; 2) performing operator splitting over each of $\{x_i\}_{i=1,2}$ independently and
3) obtaining the solution \( x \) by linear combination of \( \{x_i\}_{i=1,2} \). We call it Composite Splitting Denoising (CSD) method, which is outlined in Algorithm 6. Its validity is guaranteed by the following theorem:

**Theorem 5.2.2** Suppose \( \{x^j\} \) the sequence generated by the CSD. Then, \( x^j \) will converge to \( \text{prox}_{\rho}(\alpha \|x\|_{TV} + \beta \|\Phi x\|_1)(x_g) \), which means that we have \( x^j \to \text{prox}_{\rho}(\alpha \|x\|_{TV} + \beta \|\Phi x\|_1)(x_g) \).

**Sketch Proof of Theorem 5.2.2:**

Consider a more general formulation:

\[
\min_{x \in \mathbb{R}^p} F(x) \equiv f(x) + \sum_{i=1}^m g_i(B_i x) \tag{5.6}
\]

where \( f \) is the loss function and \( \{g_i\}_{i=1,\ldots,m} \) are the prior models, both of which are convex functions; \( \{B_i\}_{i=1,\ldots,m} \) are orthogonal matrices.

**Proposition 5.2.1** (Theorem 3.4 in [27]): Let \( H \) be a real Hilbert space, and let \( g = \sum_{i=1}^m g_i \) in \( \Gamma_0(H) \) such that \( \text{dom}g_i \cap \text{dom}g_j \neq \emptyset \). Let \( r \in H \) and \( \{x_j\} \) be generated by the Algorithm 7. Then, \( x_j \) will converge to \( \text{prox}(g)(r) \).

The detailed proof for this proposition can be found in [27] and [26].

**Algorithm 7** (Algorithm 3.1 in [27])

\[
\begin{align*}
\text{Input: } & \rho, \{z_i\}_{i=1,\ldots,m} = r, \{w_i\}_{i=1,\ldots,m} = 1/m, \\
& \text{for } j = 1 \text{ to } J \text{ do} \\
& \text{for } i = 1 \text{ to } m \text{ do} \\
& \quad p_{i,j} = \text{prox}_{\rho}(g_i/w_i)(z_j) \\
& \text{end for} \\
& \quad p_j = \sum_{i=1}^m w_i p_{i,j} \\
& \quad q_{j+1} = z_j + q_j - x_j \\
& \quad \lambda_j \in [0,2] \\
& \text{for } i = 1 \text{ to } m \text{ do} \\
& \quad z_{i,j+1} = z_{i,j} + \lambda_j (2p_j - x_j - p_{i,j}) \\
& \text{end for} \\
& \quad x_{j+1} = x_j + \lambda_j (p_j - x_j) \\
& \text{end for}
\end{align*}
\]

Suppose that \( y_i = B_i x, s_i = B_i^T r \) and \( h_i(y_i) = m \rho g_i(B_i x) \). Because the operators \( \{B_i\}_{i=1,\ldots,m} \) are orthogonal, we can easily obtain that \( \frac{1}{2\rho} \|x - r\|^2 = \sum_{i=1}^m \frac{1}{2m \rho} \|y_i - s_i\|^2 \). The above problem is transferred to:
\[\hat{y}_i = \arg \min_{y_i} \sum_{i=1}^{m} \frac{1}{2} \|y_i - s_i\|^2 + h_i(y_i), x = B_i^T y_i, i = 1, ..., m \] (5.7)

Obviously, this problem can be solved by Algorithm 7. According to Proposition 5.2.1, we know that \(x\) will converge to \(\text{prox}(g)(r)\). Assuming \(g_1(x) = \alpha \|x\|_{TV}, g_2(x) = \beta \|x\|_1, m = 2, w_1 = w_2 = 1/2\) and \(\lambda_j = 1\), we obtain the proposed CSD algorithm. \(x\) will converge to \(\text{prox}(g)(r)\), where \(g = g_1 + g_2 = \alpha \|x\|_{TV} + \beta \|\Phi x\|_1\).

**End of Proof**

Combining the CSD with FISTA, a new algorithm FCSA is proposed for MR image reconstruction problem (5.1). In practice, we found that a small iteration number \(J\) in the CSD is enough for the FCSA to obtain good reconstruction results. Especially, it is set as 1 in our algorithm. Numerous experimental results in the next section will show that it is good enough for real MR image reconstruction.

Algorithm 9 outlines the proposed FCSA. In this algorithm, if we remove the acceleration step by setting \(t^{k+1} = 1\) in each iteration, we will obtain the Composite Splitting Algorithm (CSA), which is outlined in Algorithm 8. A key feature of the FCSA is its fast convergence performance borrowed from the FISTA. From Theorem 5.2.1, we know that the FISTA can obtain an \(\epsilon\)-optimal solution in \(O(1/\sqrt{\epsilon})\) iterations.

Another key feature of the FCSA is that the cost of each iteration is \(O(p \log(p))\), as confirmed by the following observations. The step 4, 6 and 7 only involve adding vectors or scalars, thus cost only \(O(p)\) or \(O(1)\). In step 1, \(\nabla f(r^k) = R^T (R r^k - b)\) since \(f(r^k) = \frac{1}{2} \|R r^k - b\|^2\) in this case. Thus, this step only costs \(O(p \log(p))\). As introduced above, the step \(x^k = \text{prox}_\rho(2\alpha \|x\|_{TV})(x_g)\) can be computed quickly with cost \(O(p)\) [11]: The step \(x^k = \text{prox}_\rho(2\beta \|\Phi x\|_1)(x_g)\) has a close form solution and can be computed with cost \(O(p \log(p))\). In the step \(x^k = \text{project}(x^k, [l, u])\), the function \(x = \text{project}(x, [l, u])\) is defined as: 1) \(x = x\) if \(l \leq x \leq u\); 2) \(x = l\) if \(x < u\); and 3) \(x = u\) if \(x > u\), where \([l, u]\) is the range of \(x\). For example, in the case of MR image reconstruction, we can let \(l = 0\) and \(u = 255\) for 8-bit gray MR images. This step costs \(O(p)\). Thus, the total cost of each iteration in the FCSA is \(O(p \log(p))\).

With these two key features, the FCSA efficiently solves the MR image reconstruction problem (5.1) and obtains better reconstruction results in terms of both the reconstruction accuracy and computation complexity. The experimental results in the next section demonstrate its superior performance compared with all previous methods for compressed MR image reconstruction.
Algorithm 8 CSA

Input: $\rho = 1/L$, $\alpha$, $\beta$, $t^1 = 1$, $x^0 = r^1$
for $k = 1$ to $K$ do
    $x_g = r^k - \rho \nabla f(r^k)$
    $x_1 = \text{prox}_\rho(2\alpha\|x\|_{TV})(x_g)$
    $x_2 = \text{prox}_\rho(2\beta\|\Phi x\|_1)(x_g)$
    $x^k = (x_1 + x_2)/2$
    $x^{k+1} = x^k$
end for

Algorithm 9 FCSA

Input: $\rho = 1/L$, $\alpha$, $\beta$, $t^1 = 1$, $x^0 = r^1$
for $k = 1$ to $K$ do
    $x_g = r^k - \rho \nabla f(r^k)$
    $x_1 = \text{prox}_\rho(2\alpha\|x\|_{TV})(x_g)$
    $x_2 = \text{prox}_\rho(2\beta\|\Phi x\|_1)(x_g)$
    $x^k = (x_1 + x_2)/2$
    $x^{k+1} = \text{project}(x^k, [l, u])$
    $t^{k+1} = (1 + \sqrt{1 + 4(t^k)^2})/2$
    $r^{k+1} = x^k + ((t^k - 1)/t^{k+1})(x^k - x^{k-1})$
end for

5.3 Experiments

5.3.1 Experiment Setup

Suppose a MR image $x$ has $n$ pixels, the partial Fourier transform $R$ in problem (5.1) consists of $m$ rows of a $n \times n$ matrix corresponding to the full 2D discrete Fourier transform. The $m$ selected rows correspond to the acquired $b$. The sampling ratio is defined as $m/n$. The scanning duration is shorter if the sampling ratio is smaller. In MR imaging, we have certain freedom to select rows, which correspond to certain frequencies. In the following experiments, we select the corresponding frequencies according to the following manner. In the k-space, we randomly obtain more samples in low frequencies and less samples in higher frequencies. This sampling
scheme has been widely used for compressed MR image reconstruction [83, 85, 143]. Practically, the sampling scheme and speed in MR imaging also depend on the physical and physiological limitations [83].

We implement our CSA and FCSA for problem (5.1) and apply them on 2D real MR images. All experiments are conducted on a 2.4GHz PC in Matlab environment. We compare the CSA and FCSA with the classic MR image reconstruction method based on the CG [83]. We also compare them with two of the fastest MR image reconstruction methods, TVCMRI \(^1\) [85] and RecPF \(^2\) [143]. For fair comparisons, we download the codes from their websites and carefully follow their experiment setup. For example, the observation measurement \(b\) is synthesized as \(b = Rx + n\), where \(n\) is the Gaussian white noise with standard deviation \(\sigma = 0.01\). The regularization parameter \(\alpha\) and \(\beta\) are set as 0.001 and 0.035. \(R\) and \(b\) are given as inputs, and \(x\) is the unknown target. For quantitative evaluation, the Signal-to-Noise Ratio (SNR) is computed for each reconstruction result. Let \(x_0\) be the original image and \(x\) a reconstructed image, the SNR is computed as: 

\[
SNR = 10 \log_{10} \left( \frac{V_s}{V_n} \right)
\]

where \(V_n\) is the Mean Square Error between the original image \(x_0\) and the reconstructed image \(x\); \(V_s = \text{var}(x_0)\) denotes the power level of the original image where \(\text{var}(x_0)\) denotes the variance of the values in \(x_0\).

5.3.2 Visual Comparisons

We apply all methods on four 2D MR images: cardiac, brain, chest and artery respectively. Figure 5.4 shows these images. For convenience, they have the same size of \(256 \times 256\). The sample ratio is set to be approximately 20%. To perform fair comparisons, all methods run 50 iterations except that the CG runs only 8 iterations due to its higher computational complexity.

Figure 5.5, 5.6, 5.7 and 5.8 show the visual comparisons of the reconstructed results by

\(^1\)http://www.columbia.edu/~sm2756/TVCMRI.htm
\(^2\)http://www.caam.rice.edu/~optimization/L_1/RecPF/
Figure 5.5: Cardiac MR image reconstruction from 20% sampling. (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 9.86, 14.43, 15.20, 16.46 and 17.57 (db). Their CPU time are 2.87, 3.14, 3.07, 2.22 and 2.29 (s).
Figure 5.6: Brain MR image reconstruction from 20% sampling (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 8.71, 12.12, 12.40, 18.68 and 20.35 (db). Their CPU time are 2.75, 3.03, 3.00, 2.22 and 2.20 (s).
Figure 5.7: Chest MR image reconstruction from 20% sampling (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 11.80, 15.06, 15.37, 16.53 and 16.07 (db). Their CPU time are 2.95, 3.03, 3.00, 2.29 and 2.234 (s).
Figure 5.8: Artery MR image reconstruction from 20% sampling. (a) Original image; (b), (c), (d), (e) and (f) are the reconstructed images by the CG [83], TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 11.73, 15.49, 16.05, 22.27 and 23.70 (db). Their CPU time are 2.78, 3.06, 3.20, 2.22 and 2.20 (s).
different methods. The FCSA always obtains the best visual effects on all MR images in
less CPU time. The CSA is always inferior to the FCSA, which shows the effectiveness
of acceleration steps in the FCSA for the MR image reconstruction. The classical CG [83]
is far worse than others because of its higher cost in each iteration, the RecPF is slightly better than
the TVCMRI, which is consistent with observations in [85] and [143].

In our experiments, these methods have also been applied on the test images with the
sample ratio set to 100%. We observed that all methods obtain almost the same reconstruction
results, with SNR 64.8, after sufficient iterations. This was to be expected, since all methods
are essentially solving the same formulation.

5.3.3 CPU Time and SNRs

Figure 5.9: Performance comparisons (CPU-Time vs. SNR) on different MR images: a) Cardiac
image; (b) Brain image; (c) Chest image and (d) Artery image.

Figure 5.9 shows the performance comparisons between different methods in terms of the
CPU time over the SNR. Table 5.1 and 5.2 tabulate the SNR and CPU Time by different
methods, averaged over 100 runs for each experiment, respectively. The FCSA always obtains the best reconstruction results on all MR images by achieving the highest SNR in less CPU time. The CSA is always inferior to the FCSA, which shows the effectiveness of acceleration steps in the FCSA for the MR image reconstruction. While the classical CG [83] is far worse than others because of its higher cost in each iteration, the RecPF is slightly better than the TVCMRI, which is consistent to observations in [85] and [143].

5.3.4 Sample Ratios

To test the efficiency of the proposed method, we further perform experiments on a full body MR image with size of $924 \times 208$. Each algorithm runs 50 iterations. Since we have shown that the CG method is far less efficient than other methods, we will not include it in this experiment. The sample ratio is set to be approximately 25%. To reduce the randomness, we run each experiment 100 times for each parameter setting of each method. The examples of the original and recovered images by different algorithms are shown in Figure 5.10. From there, we can observe that the results obtained by the FCSA are not only visually better, but also superior in terms of both the SNR and CPU time.

To evaluate the reconstruction performance with different sampling ratio, we use sampling ratio 36%, 25% and 20% to obtain the measurement $b$ respectively. Different methods are then used to perform reconstruction. To reduce the randomness, we run each experiments 100 times for each parameter setting of each method. The SNR and CPU time are traced in each iteration.

Table 5.1: Comparison of the SNR (db) over 100 runs

<table>
<thead>
<tr>
<th></th>
<th>CG</th>
<th>TVCMRI</th>
<th>RecPF</th>
<th>CSA</th>
<th>FCSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardiac</td>
<td>12.43±1.53</td>
<td>17.54±0.94</td>
<td>17.79±2.33</td>
<td>18.41±0.73</td>
<td>19.26±0.78</td>
</tr>
<tr>
<td>Brain</td>
<td>10.33±1.63</td>
<td>14.11±0.34</td>
<td>14.39±2.17</td>
<td>15.25±0.23</td>
<td>15.86±0.22</td>
</tr>
<tr>
<td>Chest</td>
<td>12.83±2.05</td>
<td>16.97±0.32</td>
<td>17.03±2.36</td>
<td>17.10±0.31</td>
<td>17.58±0.32</td>
</tr>
<tr>
<td>Artery</td>
<td>13.74±2.28</td>
<td>18.39±0.47</td>
<td>19.30±2.55</td>
<td>22.03±0.18</td>
<td>23.50±0.20</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of the CPU Time (s) over 100 runs

<table>
<thead>
<tr>
<th></th>
<th>CG</th>
<th>TVCMRI</th>
<th>RecPF</th>
<th>CSA</th>
<th>FCSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardiac</td>
<td>2.82±0.16</td>
<td>3.16±0.10</td>
<td>2.97±0.12</td>
<td>2.27±0.08</td>
<td>2.30±0.08</td>
</tr>
<tr>
<td>Brain</td>
<td>2.81±0.15</td>
<td>3.12±0.15</td>
<td>2.95±0.10</td>
<td>2.27±0.12</td>
<td>2.31±0.13</td>
</tr>
<tr>
<td>Chest</td>
<td>2.79±0.16</td>
<td>3.00±0.11</td>
<td>2.89±0.07</td>
<td>2.21±0.06</td>
<td>2.26±0.07</td>
</tr>
<tr>
<td>Artery</td>
<td>2.81±0.17</td>
<td>3.04±0.13</td>
<td>2.94±0.09</td>
<td>2.22±0.07</td>
<td>2.27±0.13</td>
</tr>
</tbody>
</table>
Figure 5.10: Full Body MR image reconstruction from 25% sampling (a) Original image; (b), (c), (d) and (e) are the reconstructed images by the TVCMRI [85], RecPF [143], CSA and FCSA. Their SNR are 12.56, 13.06, 18.21 and 19.45 (db). Their CPU time are 12.57, 11.14, 10.20 and 10.64 (s).

Figure 5.11 gives the performance comparisons between different methods in terms of the CPU time and SNR when the sampling ratios are 36%, 25% and 20% respectively. The reconstruction results produced by the FCSA are far better than those produced by the CG, TVCMRI and RecPF. The reconstruction performance of the FCSA is always the best in terms of both the reconstruction accuracy and the computational complexity, which further demonstrates the effectiveness and efficiency of the FCSA for the compressed MR image construction.
Figure 5.11: Performance comparison on the full body MR image with different sampling ratios. The sample ratios are: (1) 36%; (2) 25% and (3) 20%. The performance: (a) Iterations vs. SNR and (b) Iterations vs. CPU Time (s).
5.3.5 Discussion

The experimental results reported above validate the effectiveness and efficiency of the proposed composite splitting algorithms for compressed MR image reconstruction. Our main contributions are:

- We propose an efficient algorithm (FCSA) to reconstruct the compressed MR images. It minimizes a linear combination of three terms corresponding to a least square data fitting, total variation (TV) and $L_1$ norm regularization. The computational complexity of the FCSA is $O(p \log(p))$ in each iteration ($p$ is the pixel number in reconstructed image). It also has fast convergence properties. It has been shown to significantly outperform the classic CG methods [83] and two state-of-the-art methods (TVCMRI [85] and RecPF [143]) in terms of both accuracy and complexity.

- The step size in the FCSA is designed according to the inverse of the Lipschitz constant $L_f$. Actually, using larger values is known to be a way of obtaining faster versions of the algorithm [142]. Future work will study the combination of this technique with the CSD or FCSA, which is expected to further accelerate the optimization for this kind of problems.

- The proposed methods are developed to efficiently solve the compressed MR image reconstruction problem, which has been addressed by SparseMRI, TVCMRI and RecPF. Therefore, with enough iterations, SparseMRI, TVCMRI and RecPF will obtain the same solution as that obtained by our methods. Since all of them solve the same formulation, they will lead to the same gain in information content. In our future work, we will develop new effective models for compressed MR image reconstruction, which can lead to more information gains.

5.4 Summary

We have proposed an efficient algorithm for the compressed MR image reconstruction. Our work has the following contributions. First, the proposed FCSA can efficiently solve a composite regularization problem including both TV term and $L_1$ norm term, which can be easily extended to other medical image applications [63]. Second, the computational complexity of the FCSA is only $O(p \log(p))$ in each iteration where $p$ is the pixel number of the reconstructed image. It also has strong convergence properties. These properties make the real compressed MR image
reconstruction much more feasible than before. Finally, we conduct numerous experiments to compare different reconstruction methods. Our method is shown to significantly outperform the classical methods and two of the fastest methods so far in terms of both accuracy and complexity.
Chapter 6

Applications in Image Processing and Computer-aided Diagnosis

This chapter introduces how to use the sparsity techniques to practical applications in image processing and computer-aided diagnosis. We will detail the sparsity related algorithms in three different applications: Tag separation in cardiac tagged MRI, Cervigram image segmentation, Transformation-invariant sparse representation for robust face recognition and dynamic scene registration.

6.1 Tag Separation in Cardiac Tagged MRI

This section introduces a tag separation method for better cardiac boundary segmentation and tag tracking, which was published in [61]. Our approach is based on two observations in the cardiac tagged MR images: 1) the tag patterns have a regular texture in tagged MR images; 2) the cardiac images without tag patterns are piecewise smooth with sparse gradients. These observations motivate us to use two dictionaries, one based on the Discrete Cosine Transform for representing tag patterns and the other based on the Wavelet Transform for representing the underlying cardiac image without tag patterns. The two dictionaries are built such that they can lead to sparse representations of the tag patterns and of the piece-wise smooth regions without tag patterns. With the two dictionaries, a new tag separation approach is proposed to simultaneously optimize with respect to the two sparse representations, where optimization is directed by the Total Variation regularization scheme. While previous methods have focused on tag removal, our approach decomposes a tagged cardiac image into a tag-only image and a tag-free image, which could be applied to to accurate tag tracking and cardiac boundary segmentation. We demonstrate the superior performance of the proposed approach through extensive experiments on large sets of cardiac tagged MR images.
6.1.1 Problem

Cross-sectional imaging techniques such as ultrasound, CT, conventional MRI, and tagged MRI provide a non-intrusive approach to obtain tomographic images of the heart at different stages in the cardiac cycle. For instance, a typical 4D spatial-temporal MRI data set has cross-sectional images of the heart at 18 locations over 24 time points in the cardiac cycle. Given such large amount of data, quantitative analysis of the heart wall motion using these images becomes attractive. Systematic quantitative analysis of heart-wall motion and blood flow often requires accurate localization and segmentation of the epicardial and endocardial surfaces from large groups of population with high efficiency. Since manual drawing of cardiac contours is very time-consuming, there has been vigorous research on methods to automate the cardiac segmentation [51][103][3].

The tagged MRI technique generates a set of tag patterns within the myocardium as temporary markers at end-diastole by spatial modulation of the magnetization. These tagged patterns appear initially in darker colors in the tagged MR images. Generally, they will persist for a short period of time in the myocardium and deform with the underlying tissue during the cardiac cycle in vivo, which provides detailed myocardial motion information. Sample images are shown in Figure 6.1. Although the added tagged patterns can facilitate the visualization of detailed transmural motion, they also pose great challenges to the cardiac image processing and analysis tasks, such as cardiac boundary segmentation.

To address this problem, several approaches have been proposed to remove the tag patterns before performing segmentation. In [51][95], morphological operations are used to fill in the region between removed tagging lines. However, it is difficult to choose reasonable parameters for a large set of tagged MR images. In [103][91], band-pass filtering approaches are proposed to enhance the tag-patterned region and increase the blood-to-myocardium contrast. However, the above filtering-based approaches greatly depend on the designed filters. They often do not adapt well to large sets of images captured under different conditions. To avoid these problems, a band-stop filtering based method was recently proposed for tag removal [105]. The images are processed in the spectral domain. Based on observations from large sets of MR images with tagging lines, they found that, except for the low frequency peak at the origin, other harmonic energy peaks placed symmetrically around the low frequency peak may be introduced by the spatial tag patterns (as shown in Figure 6.1 (b, d)). According to this observation, the optimal band-stop filter based on the Mean Shift algorithm is designed to remove these harmonic energy peaks while keeping other frequency components unchanged as much as possible. Experimental
results on 150 cardiac images with tagging lines demonstrate the superior performance over previous approaches.

![Tagged cardiac MR images and their corresponding magnitude spectrums](image)

Figure 6.1: Tagged cardiac MR images and their corresponding magnitude spectrums

However, this method still has drawbacks: 1) although it can localize the harmonic frequency regions which correspond to the spatial tagging lines, it abandons all frequency information in the localized regions, which include not only the tagging-line related frequency information but also those related to the piecewise smooth images without tagging lines. Thus, it inevitably causes some artifacts in the recovered images after tag removal; 2) when there are only a few harmonic regions, these artifacts will not be obvious since they only remove relatively less distinctive information from piecewise smooth images. However, the artifacts will be more serious when there are more harmonic regions corresponding to the spatial tag patterns in the tagged images. Thus, simply removing all frequency information in these regions will rapidly degrade the tag removal performance; 3) as all previous methods, it does not show the potential for better tag tracking, due to the lack of recovering the tag-only image.

We develop a tag separation approach according to the image decomposition theory based on sparse representations. We observe that the tag pattern appearance in the cardiac tagged MR images is approximately a regular texture, and that without the tag patterns, the cardiac images are piecewise smooth with sparse gradients. These observations motive us to seek sparse representations for both the tag patterns and the underlying piece-wise smooth image region. Two dictionaries are found. One is based on the Discrete Cosine Transform (DCT) for representing the regular tag-pattern texture, and the other is based on the Wavelet Transform (WT) for representing the image without tags. Using these two dictionaries, the tag patterns are separated from the original tagged images by jointly optimizing over the separated tag patterns and the remaining piecewise smooth regions, under the supervision of the Total Variation regularization scheme.

Different from previous tag removal methods, our tag separation approach can optimally
decompose the tagged image into one tag-only image and one remaining piecewise-smooth cardiac image. The tag-only image can then be used for tag tracking, and the remaining piecewise smooth image without tag patterns can be used for accurately localizing the cardiac boundaries. We demonstrate the superior performance of the proposed method through extensive experiments on large sets of cardiac tagged MR images, and show that the method outperforms previous methods.

![Figure 6.2: A cardiac tagged MR image can be decomposed into a tag-only image and a piecewise-smooth cardiac image without tag.](image)

### 6.1.2 Formulation

If we know the tag patterns exactly in the tagged cardiac MR images, the detailed myocardial motion information can be obtained by tracking these known tag patterns in the image sequences. If we know the piecewise smooth cardiac images without tag patterns, we can easily segment the cardiac boundary by known techniques, such as the Metamorphs model [67]. However, we only know their combination when given the tagged cardiac MR images. Therefore, it is important to decompose a tagged MR image into two separate parts: the tags, and the piecewise smooth image without the tags.

We assume that a known tagged cardiac MR image $I$ can be represented as the combination of a piecewise smooth cardiac image $S$ without tags and the tag-only image $T$:

$$I = S + T$$  \hspace{1cm} (6.1)$$

where the images $I$, $S$ and $T$ are of the same size $n \times n$. They are then reordered as 1D vectors with size $n^2 \times 1$. The decomposition problem is a typical under-constrained problem. It is impossible to obtain good solutions for $S$ and $T$ only given $I$ without prior information. Fortunately, two priors are observed from large sets of tagged cardiac MR images: 1) the tag patterns appear as a kind of regular texture in the tagged cardiac MR images; 2) the cardiac images without tag patterns are piecewise smooth with sparse gradients. Previous work has established
that the piecewise smooth images can be sparsely represented by the undecimated version of the Wavelet Transforms (UWT) while the DCT is appropriate for sparsely representing the textures [119]. Motivated by the two observed priors, we build two appropriate over-complete dictionaries $A \in \mathbb{R}^{n^2 \times m}$ and $B \in \mathbb{R}^{n^2 \times m}$, where typically $m \gg n^2$. While the dictionary $A$ is based on the DCT for representing tag patterns, the dictionary $B$ is based on the UWT for representing the cardiac image without tag patterns. Dictionaries $A$ and $B$ are built such that they can lead to sparse representations over either the tag texture or the piecewise smooth image.

Therefore, the tag separation problem is to optimally decompose image $I$ into two images $S$ and $T$, where the image $S$ should be sparsely represented by dictionary $B$ and the tag image $T$ should be sparsely represented by dictionary $A$. The problem is formulated as the image decomposition problem:

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha, \beta}{\text{argmin}} (\|\alpha\|_0 + \|\beta\|_0), \quad \text{while} \quad I = A \ast \alpha + B \ast \beta \quad (6.2)$$

where $\alpha$ and $\beta$ are sparse vectors with size $m \times 1$.

### 6.1.3 Method

Although the tag separation was formulated as an image decomposition problem in Eq. 6.2, the problem in Eq. 6.2 is NP-hard. In the general case, no known procedure can correctly find the sparsest solution more efficiently than exhausting all subsets of the entries for $\alpha$ and $\beta$ [23]. Fortunately, recent developments in sparse representation provide a theorem to efficiently solve this problem [119]:

**Definition** Given a matrix $Q$, its Mutual Incoherence $M_Q$ is defined as the maximal off-diagonal entry in the absolute Gram matrix $|Q^H \ast Q|$.

**Theorem** If the image $I$ is the combination of image $T$ and image $S$, $T = A \ast \alpha$, $S = B \ast \beta$, $Q = [A, B]$, and $\|\hat{\alpha}\|_0 + \|\hat{\beta}\|_0 < 0.5 \ast (1 + 1/M_Q)$ is true, then the solution to Eq. 6.3 leads to the global optimal solution of the $L_0$ minimization problem in Eq. 6.2, and this is the desired image decomposition

This Theorem implies that, if the true $\alpha$ and $\beta$ are sparse enough, the image decomposition problem in Eq. 6.2 is equivalent to the following $L_1$ minimization problem:

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha, \beta}{\text{argmin}} (\|\alpha\|_1 + \|\beta\|_1), \quad \text{while} \quad I = A \ast \alpha + B \ast \beta \quad (6.3)$$
It is a convex optimization problem and can be easily solved. Thus, according to the above theorem, if the input image is exactly combined from one piecewise smooth image and another tag pattern image, we definitely can successfully decompose it into the two desired output images. Unfortunately, in practice, the input tagged cardiac MR images inevitably include noise. It is impractical that the input images are exactly combined from a tag pattern and a piecewise smooth image. Following the idea in [119], we approximate this problem by relaxing the constraint in Eq. 6.3 for denoising and we add the Total Variation (TV) term for an additional penalty:

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha, \beta}{\text{argmin}} (\|\alpha\|_1 + \|\beta\|_1 + \lambda \ast \|I - A \ast \alpha - B \ast \beta\|^2 + \gamma \ast \text{TV}(B \ast \beta)) \quad (6.4)$$

Here, the TV penalty term can guide the image $S$ to fit the piecewise smooth model based on the assumption that the cardiac image without tag patterns should be a piecewise smooth image with sparse gradients. This formulation can accelerate the convergence. We know that $T = A \ast \alpha$ and $S = B \ast \beta$. When they are put back into Eq. 6.4, it is further approximated as follows:

$$\hat{T}, \hat{S} = \underset{T,S}{\text{argmin}} (\|A^+ \ast T\|_1 + \|B^+ \ast S\|_1 + \lambda \ast \|I - T - S\|^2 + \gamma \ast \text{TV}(S)) \quad (6.5)$$

where $A^+$ and $B^+$ are Moore-Penrose pseudo inverses of $A$ and $B$ respectively. Their computation is relatively easy since $A$ and $B$ are two dictionaries corresponding to the DCT and the UWT, respectively. It is known that there exist fast algorithms for DCT and UWT with complexity of no more than $n^2 \log_2(n)$ while the image size is $n \times n$. The optimization to find the solution in Eq. 6.5 can be efficiently performed by the Block-coordinate Relaxation algorithm [1].

Now, the tagged cardiac image $I$ can be efficiently and optimally decomposed into the tag pattern only image $T$ and the piecewise smooth cardiac image $S$. Then, the cardiac boundary segmentation can be performed on the piecewise smooth cardiac image $S$ instead of the tagged cardiac image $I$, and the tag tracking can be performed on the tag pattern only image $T$ instead of $I$. The experimental results in the next section will demonstrate that our tag separation approach can facilitate both tasks of cardiac boundary segmentation and tag tracking and leads to more accurate results.

### 6.1.4 Experiments

The testing data are 4D spatial-temporal short-axis cardiac tagged MR images. A 1.5T Siemens MR imaging system was used to acquire the images, using an ECG-gated tagged gradient echo.
Figure 6.3: Tag separation. (a) the input image \( I \); (b) the separated tag \( T \); (c) the piecewise smooth image \( S \); (d) noise.

pulse sequence. These images are perpendicular to an axis through the center of the LV. Each set consists of 18 – 24 phases, with 5 – 7 slices (images) per phase. We collected 60 sequences and 1200 2D-images. An expert was asked to draw the endocardium contours and the tagged lines in these images for validation purposes.

In the following, we demonstrate the superior performance of the proposed tag separation approach for better cardiac boundary segmentation and tag tracking on the above tested data. Our algorithm was implemented in MATLAB and tested on a 2GHz Pentium 4 PC. The average running time was 8 seconds for an image with size 128 × 128. Figure 6.3 shows one example of our tag separation approach. Thanks to the efficient image decomposition based on sparse representation, our approach almost perfectly decomposes the input tagged image \( I \) into the tag image \( T \), the piecewise smooth cardiac image \( S \), and noise.

To compare the band-pass filtering method [91], band-stop filtering method [105] and the proposed approach, we used them on the same set of testing data. Figure 6.4 (1) shows the comparisons on the whole image in terms of visual effect. Figure 6.4 (2) and (3) show the comparisons on local regions of the image. It shows that the results by the band-stop filtering method obtain better visual effects than those by band-pass filtering method. The former are relatively consistent with the original image appearance and can be used to recover more detailed structures. It is consistent with that presented in [105]: the band-stop filtering method can obtain superior tag removal performance compared to the band-pass filtering method [91].

With further observations in Figure 6.4 (1c), we can find that there are some artifacts in the recovered images by the band-stop filtering method. These artifacts are more when zooming in to focus on only the local regions, as shown in Figure 6.4 (2c) and (3c). This confirms a limitation of the band-stop filtering method: it can easily cause artifacts in the recovered images, since it simply abandons all frequency information in the localized harmonic frequency regions. Thanks to the efficient joint optimization and denoising schemes, our approach can recover clear
results as shown in Figure 6.4 (d). These recovered images adaptively remove the noise and only include more important structure information for the following segmentation according to the effective denoising scheme under the supervision of the Total Variation regularization.

Note that the purpose of the tag separation is to obtain more accurate cardiac boundary segmentation and better tag tracking. Therefore, we applied the Metamorphs model [67] to segment the LV endocardium on the cardiac images without tag patterns, which are recovered by different methods. Figure 6.4 (4) shows one segmentation example. It is obvious that the estimated LV endocardium contours by our approach are closer to the true boundary than those obtained by the other two methods, while the results by the band-stop filtering method are better than those obtained by the band-pass filtering method. Quantitative validation is
performed by comparing the segmentation results with the expert solution. The above experimental results show that our approach and the band-stop filtering method are far better than the band-pass filtering method [91] for tag removal. We therefore only focus on the quantitative comparison between our approach and the band-stop filtering method [105]. Denote the expert segmentation in the images as \( l_{true} \), and the results from the recovered images as \( l_{est} \). We define the False Positive Fraction (FPF) to indicate the amount of tissue falsely identified by the estimation as a fraction of the total amount of tissue in the true segmentation: \( \frac{|l_{est} - l_{true}|}{l_{true}} \). The Positive Fraction (TPF) describes the fraction of the total amount of tissue in the true segmentation that is overlapped with the estimation: \( \frac{|l_{est} \cap l_{true}|}{l_{true}} \). The quantitative comparison result is: the band-stop filtering method gained FPF = 19.6% and TPF = 83.8%, while our approach achieved FPF = 9.3% and TPF = 91.7%. The proposed approach thus outperforms the band-stop filtering method in endocardium segmentation.

To our knowledge, our method is the first to optimally separate the tag patterns for tracking using image decomposition and sparse representations. For comparisons, we use a set of snakes to track the tags in the original tagged image \( I \) and the separated tag only image \( T \) respectively. Figure 6.5 (a) and (c) show the tracking results in the original tagged images. Some mis-tracks occur in the regions indicated by green circles. Figure 6.5 (b) and (d) show the corresponding tracking results in the separated tag only images by our approach. The new results based on \( T \) by the proposed approach are better than those obtained on the original image \( I \). We further test the average and maximum tracking error in the different periods, such as Middle of Systole (MS), End of Systole (ES), Middle of Diastole (MD) and End of Diastole (ED). Table 6.1 tabulates the comparison on one example tagged image sequence. Quantitative comparisons of tracking results are also conducted on all cardiac image sequences. Denote tracked motions by the expert as \( m_{true} \), and the tracked results by snakes as \( m_{track} \). We define the Correct Tracked Rate (CTR) of the motions as: \( \frac{|m_{track} - m_{true}|}{|m_{true}|} \). While the CTR on all testing sequences is 72.8%, the CTR on all corresponding tag-only sequences (separated by our approach) is 92.3%. The above experimental results show that our approach can facilitate better tag tracking.

### 6.1.5 Summary

We have presented a tag separation approach for tagged cardiac MR images. Our approach has the following contributions. First, we introduce one self-adaptive image decomposition method for tag separation, which can be easily extended to other medical image applications. Second, the joint optimization framework plus denoising makes our approach very effective for tag separation task. Third, it is unique that our approach can optimally recover not only the
<table>
<thead>
<tr>
<th>Tracking error (pixel)</th>
<th>MS</th>
<th>ES</th>
<th>MD</th>
<th>ED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old (Avg/Max)</td>
<td>0.0897/4.2734</td>
<td>0.1880/4.3155</td>
<td>0.2311/3.9458</td>
<td>0.1497/3.9586</td>
</tr>
<tr>
<td>Proposed (Avg/Max)</td>
<td>0.0302/0.9387</td>
<td>0.0378/0.9780</td>
<td>0.0397/0.9629</td>
<td>0.0362/0.9623</td>
</tr>
</tbody>
</table>

Table 6.1: The average/maximum tracking error (pixels)

cardiac image without tag patterns for accurate cardiac boundary segmentation, but also the tag only images for better tag tracking. The potential of our approach is demonstrated through extensive experiments using large sets of tagged cardiac MR images. Our method is also shown to outperform several previous methods on tag separation, boundary segmentation and tag tracking.

![Figure 6.5: Tag tracking comparisons. (a) and (c) mis-track occurs in the areas indicated by the green circles in the original tagged image sequence; (b) and (d) the corresponding results by our approach.](image)

6.2 Cervigram Image Segmentation

We proposed an approach based on reconstructive sparse representations to segment tissues in optical images of the uterine cervix. Because of large variations in image appearance caused by the changing of the illumination and specular reflection, the color and texture features in optical images often overlap with each other and are not linearly separable. By leveraging sparse representations the data can be transformed to higher dimensions with sparse constraints and become more separated. K-SVD algorithm is employed to find sparse representations and corresponding dictionaries. The data can be reconstructed from its sparse representations and positive and/or negative dictionaries. Classification can be achieved based on comparing the reconstructive errors. In the experiments we applied our method to automatically segment the biomarker AcetoWhite (AW) regions in an archive of 60,000 images of the uterine cervix. Compared with other general methods, our approach showed lower space and time complexity and higher sensitivity.
6.2.1 Problem

Segmentation of different regions of medical images can assist doctors to analyze them. Area information from segmentation is important in many clinical cases. In this work, we propose an approach to automatically segment the biomarker AcetoWhite (AW) regions in an archive of 60,000 images of the uterine cervix. These images are optical cervigram images acquired by Cervicography using specially-designed cameras for visual screening of the cervix (Figure 6.6). They were collected from the National Cancer Institute (NCI) Guanacaste project [71] for the study of visual features correlated to the development of precancerous lesions. The most important observation in a cervigram image is the AW region, which is caused by whitening of potentially malignant regions of the cervix epithelium, following application of acetic acid to the cervix surface. Since the texture, size and location of AW regions have been shown to correlate with the pathologic grade of disease severity, accurate identification and segmentation of AW regions in cervigrams have significant implications for diagnosis and grading of cervical lesions. However, accurate tissue segmentation in cervigrams is a challenging problem because of large variations in the image appearance caused by the changing of illumination and specular reflection in pathology. As a result, the color and texture features in optical images often overlap with each other and are not linearly separable when training samples are larger than a certain level (Figure 6.7).

Previous work on cervigram segmentation has reported limited success using K-means clustering [133], Gaussian Mixture Models [47], Support Vector Machine (SVM) classifiers [68]. Shape priors are also proposed [46]. Although such priors are applicable to cervix boundary, it does not work well with AW since AW regions may have arbitrary shapes. Supervised learning based segmentation [117], [110] holds promise, especially with increasing number of features. However, because of the intrinsic diversity between images and the overlap between feature distributions of different classes, it is difficult to learn a single classifier that can perform tissue classification with low error rate for a large image set. Our empirical evaluation shows that overfitting is a serious problem when training a single (SVM) classifier using all training samples the average sensitivity of the classifier is relatively low. A potential solution is to use a Multiple Classifier System (MCS) [4], which trains a set of diverse classifiers that disagree on their predictions and effectively combines the predictions in order to reduce classification error. Voting, AdaBoost, bagging and STAPLE [138] can be employed. A necessary condition for the above ensemble methods is that all the base classifiers should provide sufficiently good performance, usually 50% or higher sensitivity and specificity in order to support the ensemble.
Figure 6.6: Examples of digitized cervicographic images (i.e. cervigrams) of the uterine cervix, created by the National Library of Medicine (NLM) and the National Cancer Institute (NCI). Ground truth boundary markings by 20 medical experts. Our work is aimed for automatically segmenting the biomarker AcetoWhite (AW) regions, which indicates clinical significance.

Figure 6.7: RGB Color distribution of AW (red) and non-AW (blue) regions. The left one comes from one sample. The right one comes from one hundred samples.

However, there may be large variance in base classifier performance in our case. Some classifiers commonly have lower sensitivity than 50%. Wang and Huang [136] proposed a method to find the best base classifier based on distance guided selection, which achieves state-of-the-art results in a subset of the archive.

In our method we focus on finding a single classifier by transforming the data to a higher dimension with sparse constraints. Then the data can be more separated using sparse representations. This classifier is potentially useful for MCS since the sensitivity and specificity are always larger than 50% in our experiments. Finding the sparse representation typically consists of the sparse coding and codebook update. Greedy algorithms such as matching pursuit (MP) and orthogonal matching pursuit (OMP) can be employed for finding sparse coefficients (coding). Extensive study of these algorithms shows that if the sought solution is sparse enough, these greedy techniques can obtain the optimal solution [129]. When the sparse data has group clustering trend, AdaDGS [60] can be employed to further improve the performance. To update
codebook, method of optimal direction (MOD) [38] and K-SVD [2] are two effective approaches. Although both of them result in similar results, we prefer K-SVD because of its better convergence rate. After finding positive and negative dictionaries from training images and computing sparse coefficients from testing images, reconstructive errors can be obtained and compared. The pixel can be assigned to the class with lower errors. Our main contributions are the following: 1) we introduce the reconstructive sparse representations and K-SVD algorithms to the medical imaging community, which are originated from the compressive sensing field; 2) we apply this theory to solve the challenging cervigram image segmentation problem and achieve improved performance.

6.2.2 Method

Here is the algorithm framework. In the training stage, ground truth is manually obtained by clinical experts. Patches on the ground truth are labeled as positive ones, while the others are negative ones. These patches are fed into K-SVD to generate positive and negative dictionaries. Then using these two dictionaries, the sparse coding step is applied on patches extracted from testing images to compute two sets of sparse coefficients. From the coefficients and corresponding dictionaries, reconstructive errors are calculated and compared for classification. An alternative way is to classify the sparse coefficients (using SVM) instead of classifying the original data, which is also tested in our experiments.

Reconstructive sparse representation is used here to classify image patches. The objective of sparse representation is to find $D$ and $X$ by minimizing the following equation:

$$\min_{D, X} \{ \| Y - DX \|_F^2 \} \text{ subject to } \forall i, \| x_i \|_0 \leq L \quad (6.6)$$

Where $Y$ represents signals (image patches here), $D$ is the overcomplete dictionary, $X$ is the sparse coefficients, $\| \cdot \|_0$ is the $l^0$ norm counting the nonzero entries of a vector, $\| \cdot \|_F$ is the Frobenius norm. Denote $y_i$ as the $i$th column of $Y$, $x_i$ as the $i$th column of $X$, then $y_i$ and $x_i$ are the $i$th signal vector and coefficient vector respectively, with dimensionality $D \in \mathbb{R}^{n \times k}$, $y_i \in \mathbb{R}^n$ and $x_i \in \mathbb{R}^k$.

K-SVD algorithm starts from a random $D$ and $X$ obtained from the sparse coding stage. The sparse coding stage is based on pursuit algorithms to find the sparse coefficient $x_i$ for each signal $y_i$. OMP is employed in this stage. OMP is an iterative greedy algorithm that selects at each step the dictionary element that best correlates with the residual part of the signal.
Then it produces a new approximation by projecting the signal onto those elements already selected [129].

In the codebook update stage K-SVD employs a similar approach as K-Means to update $D$ and $X$ iteratively. In each iteration $D$ and $X$ are fixed except only one column $d_i$ and the coefficients corresponding to $d_i$ ($i$th row in $X$), denoted as $x_i^T$. The Equation 6.6 can be rewritten as

$$
\left\| Y - \sum_{j=1}^{k} d_j x_j^T \right\|_F^2 = \left\| \left( Y - \sum_{j \neq i} d_j x_j^T \right) - d_i x_i^T \right\|_F^2
$$

(6.7)

$$
= \left\| E_i - d_i x_i^T \right\|_F^2
$$

(6.8)

We need to minimize the difference between $E_i$ and $d_i x_i^T$ with fixed $E_i$, by finding alternative $d_i$ and $x_i^T$. Since SVD finds the closest rank-1 matrix that approximates $E_i$, it can be used to minimize the Equation 6.7. Assume $E_i = U \Sigma V^T$, $d_i$ is updated as the first column of $U$, which is the eigenvector corresponding to the largest eigenvalue. $x_i^T$ is updated as the first column of $V$ multiplied by $\Sigma(1,1)$.

However, the updated $x_i^T$ may not be sparse anymore. The solution is logical and easy. We just discard the zero entries corresponding to the old $x_i^T$.

Using K-SVD algorithm we can obtain two dictionaries for positive patches and negative patches separately, denoted as $D_+$ and $D_-$ respectively. The simplest strategy to use dictionaries for discrimination is to compare the errors of a new patch $y$ reconstructed by $D_+$ and $D_-$ and choose the smaller one as its type, as shown in equation 6.9.

$$
type = \arg \min_{i=+,-} \left\{ \| y - D_i x \|_2^2 \right\} \text{ subject to } \| x \|_0 \leq L
$$

(6.9)

The potential problem of this method is that the dictionaries are trained separately. That is to say, positive/negative dictionary only depends on positive/negative patches, so it attempts to reconstruct better for positive/negative patches but not worse for negative/positive patches. Discriminative methods can be considered to alleviate this problem [86]. However, the tuning parameters of the discriminative system are very sensitive and it can only converge within small intervals. In our case, reconstructive method works relatively well. Discriminative method with this application is left to future investigation.

An alternative way is to classify the sparse coefficients $x$ instead of $y$. $x$ from training images can be fed into SVM or other classifier. The intuition is that $x$ is in higher dimension with
sparse constants and can be more separated. Both of these two approaches are tested in Section 5.3.

Since there is no shape information considered, the resulting areas are usually disconnected. Inspired by the edge linking stage of Canny edge detector, similar procedure can also be applied on this application. Equation 6.9 can be rewritten as:

\[
error = \|y - D_-x\|^2 - \|y - D_+x\|^2
\]  

(6.10)

When \( error < 0 \), the testing data is assigned to the negative samples. Otherwise it is positive. However, due to noise, there may be positive instances below the threshold (0). Thus similar to the Canny edge detector, two thresholds \( T_1 \) and \( T_2 \) (\( T_1 > T_2 \)) can be predefined. In the first pass, \( T_1 = 0 \) is used as the threshold and classification is performed. In the second pass, \( T_2 < 0 \) is set as the new threshold. The errors of neighboring points of the first results are checked, and the points with \( error > T_2 \) are merged into the positive samples. With ideal thresholds, the disconnectivity problem can be alleviated in a certain level. However, the value of \( T_2 \) highly depends on the application and currently is found by cross validation and brute force. Starting from 0, \( T_2 \) is decreased by a small step each time. The sensitivity and specificity are computed in each step. The parameters causing the best performance are chosen. More sophisticated approaches are left for future investigations.

6.2.3 Experiments

The method was implemented in Matlab R2009a and tested on a 2.40 GHz Intel Core2 Quad computer with 8G RAM. It was compared with SVM and \( k \) nearest neighbors. SVM failed to handle 90,000 patches since it would consume most memories and couldn’t converge. Thus the data for SVM was down sampled. Instead of feeding image patches into SVM, we also trained SVM using sparse coefficients. Nearest neighbor method was also time and space consuming because of the large training set. K-SVD was more efficient with 5 seconds for each iteration and less than 1GB RAM because of its sparsity.

Cervigram images from the NCI/NLM archive with multiple-expert boundary markings are available for training and validation purposes. 100 images of diverse appearance were selected for training and testing. To maximally mix the samples, 10 image is used for testing and validation and the remaining 90 ones are used for training. The mean sensitivity and specificity are reported. Different color spaces including RGB, HSV and Lab are tested. HSV is chosen since it is slightly better. Other color spaces, texture and appearance information can also be considered.
Table 6.2: Performance comparison between 4 classifiers, measured by the mean of sensitivity and specificity.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM with image patches</td>
<td>50.24%</td>
<td>63.59%</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>55.62%</td>
<td>69.18%</td>
</tr>
<tr>
<td>Compare reconstructive errors</td>
<td>62.71%</td>
<td>75.85%</td>
</tr>
<tr>
<td>SVM with sparse coefficients</td>
<td>61.46%</td>
<td>76.37%</td>
</tr>
</tbody>
</table>

Each patch is a 5 by 5 square centered in the pixel and concatenated H,S,V information into single vectors (75 by 1, \( n = 75 \)). We choose the sparse factor \( L = 6 \) and dictionaries of size \( k = 256 \). Although there are many choices for these values, they are not arbitrary. They need to satisfy the constraints mentioned in [119] to guarantee convergence. In each image 1,000 patches are randomly selected from both AW and non-AW regions, 500 for each. Overall 90,000 patches are generated from the training images. 60 iterations of K-SVD are performed. The positive and negative dictionaries represent AW and non-AW regions respectively. Each column of dictionaries are reshaped as 5 by 5 patches, and they are displayed together in Figure ??.

Some patches from different dictionaries are similar, proving that classification is a challenging task. Utilizing sparse representations can alleviate this problem.

Table 6.2 shows the results of different classifiers measured by sensitivity and specificity. Figure 6.8 visualizes the segmentation results of a specific image. Since the distributions of image patches were overlapped, SVM with image patches underperformed the classification task. Using sparse coefficients in higher dimension, SVM performed better since the data was more separated. Although comparing reconstructive errors also achieved good results, it still had many noises from non-AW regions. The reason is that the K-SVD algorithm is not discriminative and there is no shape information considered.

From the comparisons in Table 6.2 and Figure 6.8 we can find that sparse representations are very useful in this challenging problem.

### 6.2.4 Discussions

In this section, we proposed classifiers based on reconstructive sparse representations to segment tissues in optical images of the uterine cervix. Our method was compared with some typical learning methods and worked better in this challenging case. In the future we would like to put efforts on three directions. 1) We can add our method into “best base classifier” [136] to improve the sensitivity and specificity of the whole system. 2) Combining sparse representations with SVM is also interesting.
Figure 6.8: Results of different algorithms applied on two images. 1st column: original image; 2nd column: ground truth; 3rd column: SVM with radial basis function kernel; 4th column: nearest neighbor; 5th column: reconstructive sparse representations.

6.3 Transformation-invariant Sparse Representation

Sparse representation is gaining increasing attention due to its success in various applications. As we demonstrate in this section, however, image sparse representation is sensitive to image plane transformations such that existing approaches can not reconstruct the sparse representation of a geometrically transformed image. We introduce a simple technique for obtaining transformation-invariant image sparse representation.

6.3.1 Problem

The sparse representation theory has shown that sparse signals can be exactly reconstructed from a small number of linear measurements [20, 21, 32, 33, 114]. It leads to the problem: given the linear measurements \( y \in \mathbb{R}^m \) of a sparse signal \( x \in \mathbb{R}^n \), \( y = Ax \), how to reconstruct the sparse signal \( x \) from its linear measurements \( y \)? Obviously, this problem can be formulated with \( L_0 \) minimization:

\[
x_0 = \text{argmin} \|x\|_0 \quad \text{while} \quad y = Ax
\]

(6.11)

where \( \| \cdot \|_0 \) denotes the \( L_0 \)-norm that counts the number of nonzero entries in a vector. This problem is NP-hard. In the general case, no known procedure can correctly find the sparsest solution more efficiently than exhausting all subsets of the entries for \( x \). Recent developments in compressive sensing [23] show that if the original signal \( x \) is sparse enough, the above \( L_0 \)-minimization problem is equivalent to the following \( L_1 \)-minimization problem:

\[
x_0 = \text{argmin} \|x\|_1 \quad \text{while} \quad \|y - Ax\|_2 < \epsilon
\]

(6.12)
where $\varepsilon$ denotes the noise level. If the solution is really sparse and has $k$ nonzero entries, it can be efficiently solved by the homotopy algorithms in $O(k^3 + n)$ time [33]. If the signal has $k$ nonzero entries, $m = O(k \ast \log(n/m))$ linear measurements are sufficient to reconstruct the original signal exactly with high probability [20, 23]. Moreover, if the signals are not exactly $k$ sparse but can be represented by $k$ of active elements as well as contaminated with noise, sparse representation theory in compressive sensing can also handle this case with random projection analysis.

A novel approach for face recognition is recently proposed based on the sparse representation [141]. The assumptions are that the training images of a single object span a subspace and that a target test image can be sparsely represented by the entire set of training images. Therefore, the face recognition problem is treated as searching for a sparse representation of a given test image. This treatment dexterously casts recognition as a globally sparse representation problem, and the sparse presentation theory in compressive sensing can then be utilized to efficiently solve it. Experimental results [141] showed that the sparse representation recognition approach achieved favorable performance compared with other state-of-the-art methods under various conditions. It further demonstrates the power of sparse representation via $l^1$ minimization. One limitation of the method, however, is that it only handles cases in which all training images and the test image are well aligned and have the same pose. Thus, all training images and testing images have to be aligned to the same pose before face recognition. While the training images can be easily aligned off-line, aligning each test image to model images is a difficult task in practical applications. It is thus desirable to develop a transformation-invariant image sparse representation to overcome the difficulty. Obviously, this limitation comes from the fact that the current image sparse representation is sensitive to image plane transformations, which cause the nonlinear measurements and make the linear sparse representation model unreasonable.

In this section, we propose a new algorithm to make sparse representation invariant to image-plane transformations. The proposed approach aims to simultaneously recover the image plane transformation and sparse representation when a test image is not aligned with the model images. It is inspired by two simple observations: 1) if the model images span a linear subspace, their transformed versions w.r.t. some group of small transformations still can span a linear subspace in a higher dimension; 2) if a transformed version of the test image, which is aligned with the model images, lives in the above subspace, the test image after applying more and more accurately estimated transformations will get gradually closer to the subspace. When the transformation between the test image and model images is small, the first observation motives us to convert a nonlinear model representation to a linear one by increasing the dimensionality.
of the model representation. However, this scheme is no longer effective in the presence of large transformations. To resolve this problem, we turn to the second observation and recent developments in random projection manifolds [9, 57], by iteratively projecting the unaligned test image to random-projection manifolds of an extended linear model. The projections can then be separated into the aligned projection target and some residue due to misalignment. The more accurate the estimated transformation parameters are, the closer the transformed version of the test image should be to the linear sparse representation. Under this framework, we can simultaneously recover the sparse representation of the target image based on the model images and the image plane transformation between the target image and the model images.

### 6.3.2 Related Work

**Convex optimization**

As introduced above, sparse solutions can be obtained by performing $l_1$ minimization instead of $L_0$ minimization [23]. Thus, efficient $L_1$ minimization becomes the core of the sparse representation problem. Recently, several efficient algorithms are developed and they only require matrix-vector products operations [33, 75, 39].

The $L_1$-magic package implements the algorithms introduced in [21, 33]. The $L_1$ minimization is recasted as a second-order cone program and then a primal log-barrier approach is applied. In the process, only multiplications by $A$ (eqn. 6.12) and its transpose are required. One $L_1$ regularization algorithm is developed for a large-scale least squares reconstruction problem [75]. A specialized interior-point method is employed to solve this problem, which uses preconditioned conjugate gradient method to approximately solve linear systems in a truncated-Newton framework. Each search step requires only multiplications by $A$ and its transpose. The proposed $l1$-ls package is reported to outperform all previous implementations for $L_1$ minimization, including the $L_1$-magic package. Gradient Projection for Sparse Reconstruction (GPSR) is another interior point approach, which considers the sparse presentation reconstruction problem as a bound-constrained quadratic program. In order to accelerate convergence, a variant of Barzilai-Borwein steps is optionally applied when the projected gradient steps are used. Their experiments show the GPSR appears to be faster than state-of-the-art algorithms, including $l1$-ls, especially in large-scale settings. Moreover, it does not require application-specific tuning. Considering these advantages, the $L_1$ minimization in our algorithm is based on their GPSR-4.0 package.
Sparse Representation

A pioneering attempt was conducted to use the sparse representation theory for face recognition and the proposed ”Randomfaces” algorithm obtained very good face recognition results [141]. We briefly review their method here.

First, each image with size \( w \times h \) is stacked as a vector \( I_{i,n_i} \in \mathbb{R}^m \), where \( i \) is the subject number and \( n_i \) is the image number of each subject. The whole training image model can be represented as follows:

\[
A = [I_{1,1}, I_{1,2}, ..., I_{1,n_1}, ..., I_{k,n_k}] \in \mathbb{R}^{m \times n} \quad (6.13)
\]

Here, \( k \) is the total number of the subjects and \( n \) is the total number of training images. Based on the assumption that the vectors of each subject span a subspace for this subject, the new test image \( y \in \mathbb{R}^m \) of subject \( i \) can be represented as a linear combination of the training images of subject \( i \):

\[
y = \sum_{j=1}^{n_i} \alpha_{i,j} I_{i,j} \quad (6.14)
\]

where \( \alpha_{i,j}, j = 1, ..., n_i \) are weights. Then the test image \( y \) of subject \( i \) can be sparsely represented in terms of all training images:

\[
y = Ax_0 \in \mathbb{R}^m \quad (6.15)
\]

where \( x_0 = [0, ..., 0, \alpha_{i,1}, ..., \alpha_{i,n_i}, 0, ..., 0] \in \mathbb{R}^n \) is a coefficient vector whose entries are zero except those associated with subject \( i \). The sparse representation is obtained if subject number \( k \) is reasonably large. The only problem is that the dimension of the data is very high. Motivated by the theoretical results in [20, 23], random projection is used to reduce the data dimension:

\[
\tilde{y} = Ry = RAx_0 = \tilde{A}x_0 \in \mathbb{R}^d \quad (6.16)
\]

where \( R \in \mathbb{R}^{d \times m} \) with \( d \ll m \) is a random projection matrix. Until now, the face recognition is dexterously formulated as the linear sparse representation problem:

\[
x_0 = \arg\min \|x\|_0 \quad \text{while} \quad \tilde{y} = \tilde{A}x \quad (6.17)
\]

As introduced above, this problem is equivalent to the \( l^1 \) minimization problem in equation 6.12, which can be efficiently solved. The remaining problem is to identify the test image \( y \) by encoding \( x_0 \) after \( L_1 \) minimization:

\[
\text{identity}(y) = \arg\min_i E[r_i], \quad E[r_i] = \frac{1}{l} \sum_{t=1}^{l} r_i^t \quad (6.18)
\]
where $r_i$ is the residual and $r_i(y) = \|\tilde{y} - \tilde{A}\delta_i(x)\|_2$. In the new vector $\delta_{i}(x)$, the entries in $x$ associated with subject $i$ keep unchanged and others are set as zeros.

Figure 6.9(c) shows several sparse representation examples by the Randomfaces algorithm [141]. The images are also from the Extended Yale B database [43]. This database consists of 2,414 frontal-face images of 38 individuals. The image size $192 \times 168$. All images are aligned and normalized. We randomly select half of the images for training (32 images per subject), and the other half for testing. In Figure 6.9, different rows represent different subjects. In each row, Column (a) shows one of training images of the subject. Column (b) shows one test image of this subject, whose sparse reconstruction based on sparse solution and the model images by Randomfaces algorithm is shown in column (c). Very good reconstruction results were obtained using their approach, which demonstrates the effectiveness of sparse representation.

![Figure 6.9: Sparse representation results on aligned images. (a) One of training images; (b) the test image; (c) the linear sparse representation result [141] and the sparse representation result using the proposed method.](image-url)
However, the current Randomfaces algorithm cannot handle the case where the test images are not aligned with the training images. In an experiment, we introduced small translations (15 pixels in both horizontal and vertical directions) to a set of test images so that they are not aligned with the model training images. Then all the images are cropped to the size $177 \times 153$. Several sparse representation results by the Randomfaces algorithm on the unaligned test images are shown in Figure 6.10(c). One can clearly see that there are ghost effects due to the misalignment between the test images and the model images, which shows that the current linear sparse image representation approach depends on correct alignments and lacks of invariance to images plane transformations.

### 6.3.3 Transform-invariant Sparse Representation

In this section, we describe a sparse representation invariant to image-plane transformations.

#### Formulation

When there exist image plane transformations between the test images and the model images, the problems (6.11) and (6.12) can be reformulated as follows:

$$
(x_0, \beta_0) = \arg \min_x \|x\|_1, \quad T(y, \beta_0) = Ax
$$

where $\beta_0$ is the parameter of the image transformation between the test image $y$ and model images $A$. Here, $A$ is assigned according to equation 6.13 and $T(y, \beta_0)$ represents the transformed version of image $y$ with parameter $\beta_0$. In this problem, given the model $A$ and the unaligned image $y$, we attempt to simultaneously recover the sparse solution $x_0$ and the image plane transformation $\beta_0$. It is a typical Chicken-and-Egg problem. If we know the exact image plane transformation, the sparse solution can be easily obtained by $L_1$ minimization in problem (6.12) just as done in a previous approach [141]. If we know the exact sparse solution, we can obtain the sparse representation according to the sparse solution, and then the image plane transformation can be easily estimated by classical motion estimation methods [15]. However, we know neither the image plane transformation nor the sparse solution. We therefore face a highly ill-posed problem.

#### Algorithm

Our task is to simultaneously recover the image plane transformation and the sparse representation while the model images are aligned but the test image is not aligned to the model images. For convenience, we use face recognition as an example to introduce our algorithm below.
Figure 6.10: Sparse representation results given unaligned test images. Training images (a), test images (b), results by Randomfaces [141] (c), results by the proposed approach.
We consider the translation transformation first. Let \( I(\mathbf{x}) \) be an image where \( \mathbf{x} = (x_1, x_2) \). \( I(\mathbf{x} + \beta) \) represents its translated version with parameter \( \beta = (a_1, a_2) \). When the transformation parameter \( \beta \) is small, we have:

\[
T(I, \beta) = I(\mathbf{x} + \beta) \approx I(\mathbf{x}) + a_1 I_{x_1} + a_2 I_{x_2}
\]

(6.20)

where \( I_{x_1} = \frac{\partial I}{\partial x_1} \) and \( I_{x_2} = \frac{\partial I}{\partial x_2} \).

Similarly, the affine transformed version of \( I \) can be represented as:

\[
T(I, \beta) \approx I(\mathbf{x}) + a_1 I_{x_1} + a_2 x_1 I_{x_1} + a_3 x_2 I_{x_1} + a_4 I_{x_2} + a_5 x_1 I_{x_2} + a_6 x_2 I_{x_2}
\]

(6.21)

where \( \beta = (a_1, a_2, a_3, a_4, a_5, a_6) \). Now, let us consider the problem in equation 6.19.

In the case of translation transformation, let \( \beta_0 = (a_1, a_2) \) represent the transformation between the model images and the test image \( y \) of subject \( i \). Then, the translated version of the test image \( y' \), with parameter \( \beta_0 \), can be represented as:

\[
y' = T(y, -\beta_0) = \sum_{j=1}^{n_i} \alpha_{i,j} I_{i,j}
\]

(6.22)

Considering that translating \( y \) toward \( y' \) is equivalent to translating the aligned model images toward \( y \), we can rewrite the above equation:

\[
y = T(y', \beta_0) = \sum_{j=1}^{n_i} \alpha_{i,j} T(I_{i,j}, \beta_0) = \sum_{j=1}^{n_i} [\alpha_{i,j} I_{i,j} + \alpha_{i,j} a_1 I_{i,j,x_1} + \alpha_{i,j} a_2 I_{i,j,x_2}]
\]

(6.23)

where \( I_{i,j,x_1} \) is \( \frac{\partial I_{i,j}}{\partial x_1} \) and \( I_{i,j,x_2} \) is \( \frac{\partial I_{i,j}}{\partial x_2} \). The equations form a linear system:

\[
y = Bx, B = [A_0, A_1, A_2], x = [z_0, z_1, z_2]^T
\]

\[
z_0 = [0, ..., 0, \alpha_{i,1}, ..., \alpha_{i,n_i}, 0, ..., 0]
\]

\[
z_1 = [0, ..., 0, a_1 \alpha_{i,1}, ..., a_1 \alpha_{i,n_i}, 0, ..., 0]
\]

\[
z_2 = [0, ..., 0, a_2 \alpha_{i,1}, ..., a_2 \alpha_{i,n_i}, 0, ..., 0]
\]

\[
A_0 = [I_{1,1}, I_{1,2}, ..., I_{1,n_i}, ..., I_{k,n_k}]
\]

\[
A_1 = [I_{1,1,x_1}, I_{1,2,x_1}, ..., I_{1,n_i,x_1}, ..., I_{k,n_k,x_1}]
\]

\[
A_2 = [I_{1,1,x_2}, I_{1,2,x_2}, ..., I_{1,n_i,x_2}, ..., I_{k,n_k,x_2}]
\]

(6.24)

where we now obtain the linear image model \( B \in \mathbb{R}^{m \times 3n} \) for translation transformations \(^1\), as similarly done in [115]. In this way, the unaligned image \( y \) of subject \( i \) can be sparsely

\(^1\)Similarly, we can obtain \( B \in \mathbb{R}^{m \times 7n} \) for affine transformations.
represented in terms of all training images and their derivatives. The random projection is used to reduce the data dimensionality:

$$\tilde{y} = Ry = RBx = \tilde{A}x \in \mathbb{R}^d$$  \hspace{1cm} (6.25)

where $R \in \mathbb{R}^{d \times m}$ with $d \ll m$ is a random projection matrix. $l^1$ minimization instead of $l^0$ minimization is performed to derive the sparse solution:

$$x_0 = \text{argmin}\|x\|_1 \quad \text{while} \quad \|\tilde{y} - \tilde{A}x\|_2 < \epsilon$$  \hspace{1cm} (6.26)

With the computed sparse solution $x_0 = [z_0, z_1, z_2]^T$, the random projection $\tilde{y}$ can be separated into the aligned projection target $\tilde{y}_{\text{est}}$ and the residue $\tilde{y}''$:

$$Ry_{\text{est}} = \tilde{y}_{\text{est}} = \tilde{A}z_0 = RA_0z_0$$  \hspace{1cm} (6.27)

where $y_{\text{est}}'$ is the estimation of the aligned version $y'$ of the test image $y$. The recent developments [9, 57] in random projection manifolds provide the following scheme for estimating the aligned target $y'$ from equation 6.27.

**Lemma 1** Let $M$ be a compact $k$ dimensional manifold in $\mathbb{R}^m$ having volume $V$ and condition number $1/\tau$. Fix $0 < \epsilon < 1$ and $0 < \rho < 1$. Let $R$ be a random orthoprojector from $\mathbb{R}^m$ to $\mathbb{R}^d$ and

$$d \geq O\left(\frac{k \ast \log(mV\tau^{-1}) \log(\rho^{-1})}{\epsilon^2}\right)$$  \hspace{1cm} (6.28)

suppose $d < m$, then, with probability $1 - \rho$, the following statement holds: for every pair of points $x, y \in M$, and $i \in \{1, 2\}$,

$$(1 - \epsilon)\sqrt{\frac{d}{m}} \leq \frac{\|Rx - Ry\|_i}{\|x - y\|_i} \leq (1 + \epsilon)\sqrt{\frac{d}{m}}$$  \hspace{1cm} (6.29)

A fundamental connection between this Lemma and the sparse representation theory has been identified in compressive sensing [8, 9]. It states that, when the projections of two points in a random projection manifold are close, then the two original points are also close, only if these two points live in the same compact manifold. According to this, we can get:

$$\frac{\|y_{\text{est}}' - \tilde{A}z_0\|_2}{\|y_{\text{est}}' - A_0z_0\|_2} \approx \sqrt{\frac{d}{m}}$$  \hspace{1cm} (6.30)

Then, $y_{\text{est}}' \approx A_0z_0$ can be obtained from equation 6.27 and 6.30. Since $y_{\text{est}}'$ is the estimation of the aligned version $y'$ of the test image, we can optimize the translations between $y$ and $y_{\text{est}}'$ by a model based approach [15]:

$$\triangle \beta = \text{argmin}_{\beta} \|T(y, \beta) - y_{\text{est}}'\|_2$$  \hspace{1cm} (6.31)
With the estimated transformation parameters, the test image $y$ is warped towards $y_{est}$. Then, the warped image is projected again onto the manifolds defined by the model matrix $B$ and the random projection matrix; this process repeats until the residue is gradually reduced to a certain level. The complete procedure is summarized in algorithm 6.3.3.

**Algorithm 10 Transform-invariant Sparse Representation (TSR)**

1. **Input:** The training image matrix $A_0$ from $k$ subjects, a test image $y \in \mathbb{R}^m$ and iteration number $s$.
2. Build the model matrix $B = [A_0, A_1, A_2] \in \mathbb{R}^{m \times 3n}$ (For affine model, $B = [A_0, A_1, A_2, A_3, A_4, A_5, A_6] \in \mathbb{R}^{m \times 7n}$)
3. Generate $l$ random projections $R_1, ..., R_l \in \mathbb{R}^{d \times m}$.
4. **for all** $p = 1, ..., l$ **do**
   5. $\beta = 0$
   6. **for all** $q = 1, ..., s$ **do**
   7. Compute $y_{est} = T(y, -\beta)$
   8. Compute $\beta = \beta + \Delta\beta$ until $\Delta\beta$ small enough.
   9. **end for**
10. Compute $y_{est} = A_0z_0$, here $z_0 = x_0(1 : n)$.
11. Compute $\beta = \beta + \Delta\beta$ until $\Delta\beta$ small enough.
12. **end for**
13. Compute $r_i^p = \|\tilde{y} - \tilde{A}\delta_i(x_0)\|$ for $i = 1, ..., k$
14. Compute $\beta = \arg\min E[r_i]$  
15. **end for**
16. **Output:** $\text{identity}(y)$ and $\text{transform}(y)$.

It is worth noting that, the above process can also be implemented in a coarse-to-fine framework, where the procedure is applied at each level of the pyramid.

**Simultaneous Face Alignment and Recognition**

The proposed approach can be useful in face detection followed by identification, where the target image obtained by the detection module is possibly not aligned with the model images although all the model images are already aligned. In this scenario, Algorithm 6.3.3 can be directly used for simultaneous face alignment and recognition. Moreover, alignment and recognition can interact in a loop to improve each other’s performance. Better alignment leads to more accurate sparse solution, which in turn makes possible better recognition performance. On the other hand, more accurate sparse solution allows to perform better alignment.

Figure 6.9(d) shows several sparse representation examples by the proposed algorithm on aligned test images. For the first and second subjects, there are almost no differences between
the sparse representation results by the proposed approach and the Randomfaces algorithm. For the third subject, there exists slight rotation between the test image and the model images. Our result using the translation model is not perfect, but it is still better than that by Randomfaces, which produces severe ghost effects. This further confirms our conclusion: simultaneous transformation and sparse representation recovery is very important. We also tested our approach using test images that are not aligned with model images (15-pixels shift in both horizontal and vertical directions). Figure 6.10(d) shows several examples. The results are very promising and we were able to obtain both the sparse representation and the translation motion; this demonstrates that the proposed approach can generate transformation invariant sparse representation.

**Online Dynamic Texture Registration**

Online video registration is required by many video analysis applications when a video sequence is captured by a moving camera. Traditional methods generally make the brightness constancy assumption [15]: $I(x_1, x_2, t) = I(x_1, x_2, t - 1)$, where $(x_1, x_2)$ denotes the spatial coordinates and $t$ represents the time frame. However, this assumption is often violated in dynamic scenes.

Fitzgibbon [40] proposed to perform dynamic scene registration by minimizing the entropy function of an auto regressive process, which results in a difficult non-linear optimization problem. Dynamic Texture Constancy Constraint (DTCC) is introduced in [135] to solve this problem, instead of the brightness constancy. In [58], another solution is proposed by jointly optimizing over registration parameters, the average image, and the dynamic texture model according to certain prior models. These three methods involve complex optimization and do not suit well the needs of online video registration. One online video registration method proposed in [106] attempts to solve two independent subproblems: 1) the extrapolation of the preceding frames using block based video synthesis techniques; 2) the alignment of a new image frame to best fit the above extrapolation [15].

We propose a new online dynamic texture registration approach, based on the sparse representation constancy assumption instead of the traditional brightness constancy assumption. The sparse representation constancy assumption states that, given a new frame, its aligned version should be represented as a linear combination of as few preceding image frames as possible. As we know, a dynamic scene is called a dynamic texture when it is captured by a static camera and its temporal evolution exhibits certain stationarity [35]. Thus, our assumption is reasonable for dynamic-texture image sequences. Our experimental results in the next section also confirm the validity of this assumption. As a matter of fact, the traditional brightness
constancy assumption seeks that the aligned version of the current image frame can be best represented by a single preceding frame, while the proposed sparse representation constancy assumption seeks that the aligned version of the current image frame can be best represented by all preceding image frames via $l_1$ minimization. Thus, the former can be thought as a special case of the latter.

Suppose a video sequence consists of frames $I_1, ..., I_n \in \mathbb{R}^m$. Without loss of generality, we can assume that the first $k$ frames have already been aligned to the $k^{th}$ frame. Let $A_0 = [I_1, ..., I_k] \in \mathbb{R}^{m \times k}$. Considering the translation model, our task is to estimate the translation motion between the $(k+1)^{th}$ frame and the preceding frames:

$$ (x_0, \beta_0) = \arg\min ||x||_1, \quad T(y, \beta_0) = A_0 x $$

(6.32)

where $\beta$ is the motion parameter. Obviously, this problem is equivalent to the problem in equation 6.19 and can be efficiently solved by Algorithm 6.3.3. After recovering the motion $\beta$ between the $(k+1)^{th}$ frame and preceding frames, we can warp all preceding frames toward the $(k+1)^{th}$ frame according to the estimated motion parameter $\beta$. The same procedure can be applied to aligning with the $(k+2)^{th}$ frame, and so on.

For long video sequences, it is impractical to build a model matrix $A_0 = [I_1, ..., I_{t-1}] \in \mathbb{R}^{m \times (t-1)}$, where $t$ denotes the current frame number. In order to cope with this case, we can set a time window width parameter $\tau$. We then build the model matrix, $A_0 = [I_{t-\tau}, ..., I_{t-1}] \in \mathbb{R}^{m \times (t-\tau)}$, for the $t^{th}$ frame, which can avoid the memory requirement blast for a long video sequence. The complete algorithm for online dynamic texture registration is summarized below.

**Algorithm 11** TSR Based Online Dynamic Texture Registration

1: **Input:** The video sequence $I_1, ..., I_n$, the number $k$ which means $1^{st} \sim k^{th}$ have been aligned to $k^{th}$ frame, the time window width $\tau \leq k$
2: **for** all $t = k + 1, ..., n$ *do*
3: \hspace{1em} Set $A_0 = [I_{t-\tau}, ..., I_{t-1}]$
4: \hspace{1em} Set $y = I_t$ and iteration number $s$.
5: \hspace{1em} Perform Algorithm 6.3.3, $\beta_t = TSR(B, y, s)$
6: \hspace{1em} Warp $I_1, ..., I_{t-1}$ toward $I_t$ according to $\beta_t$
7: **end for**
8: **Output:** The registered $I_1, ..., I_n$ and $\beta_{k+1}, ..., \beta_n$.

### 6.3.4 Experiments

The proposed transformation invariant sparse representation is applied to face recognition and online dynamic texture registration respectively.
Face Recognition

In this section, we validate respectively the identification and verification performance of Algorithm 6.3.3 for face recognition using a public face database, namely, the Extended Yale B database [43]. This database consists of 2,414 frontal-face images of 38 individuals. The image size is 192 × 168. We randomly selected 20 subjects, half of whose images are used for training (32 images per subject), and the other half for testing. There are a total of 640 images from the 20 subjects for training. In the identification experiment, there are 640 images for testing. In the verification experiment, there are 1198 test images, half of which are true outliers.

All images are aligned and normalized in the Extended Yale B database. To evaluate the identification performance of the proposed approach, we generated shifted test images according to different shift values. For example, if the shift value is 7, each test image is shifted with random parameters between 0 and 7 pixels in both horizontal and vertical directions. The training images are kept unchanged. For fair comparison, the implementations of the Randomfaces algorithm and the proposed algorithm use the same parameters as those introduced in [141] (random projection matrix number is $l = 5$, error distortion $\varepsilon = 0.05$, and the reduced dimension $d$ is 504). Figure 6.11(a) shows the recognition performance of the proposed algorithm and the Randomfaces algorithm, as a function of shift values. One can see that the proposed algorithm outperforms the Randomfaces algorithm. When the shift value is smaller than 2 pixels, our results are slightly better than Randomfaces. When the shift value exceeds 2 pixels, the Randomfaces recognition performance is much degraded, which further demonstrates that the previous sparse image representation is sensitive to image plane transformations. In comparison, the proposed transform-invariant sparse representation achieves better and more stable recognition performance.

The verification performance of the proposed algorithm is evaluated using 1198 shifted test images with the shift value 7. Among the 1198 images, 558 images are true outliers. Since the role of verification is to reject test images that are difficult to classify, we also use the Sparse Concentration Index (SCI) [141] as an indication of confidence:

$$SCI(x_0) = \frac{k * \max_i \| \delta_i(x_0) \|_1 / \| x_0 \|_1 - 1}{k - 1} < \sigma$$ (6.33)

where $k$ is the subject number, $\sigma \in [0, 1]$ is a preselected threshold, and $\delta_i(x_0)$ represents the entries in $x_0$ associated with subject $i$ keep unchanged and others are set as zeros. We plot the Receiver Operating Characteristic (ROC) curve according to different $\sigma$ values in Figure 6.11(b). As expected, the proposed algorithm outperforms Randomfaces.
We also tested the proposed algorithm’s invariance and robustness to the affine transformation model. We scaled up the 640 test images by a factor of 1.02 and then rotated them by 2 degrees, 4, and 6, respectively. This gave us 3 groups of test images (640 in each). Table 6.3 tabulates the comparison of the identification results, which shows our algorithm has better invariance to affine transformations.

### Dynamic Scene Registration

The first set of experiments uses the Escalator sequence (shown in Figure 6.12). It includes 157 image frames. We resized each image frame to $120 \times 160$ pixels. In order to evaluate the proposed approach, we generated 3 new video sequences by transforming each image frame...
with a known motion and record the motion as ground truth. For comparison, we implemented the classic model-based motion estimation method [15], which we call Bergen’s method. Since the proposed method is an online registration method and assumes that the beginning frames have been aligned, we only compared the motion estimations from the 81st to the last frame in this experiment. The comparison results on one generated sequence are shown in Figure 6.13. The proposed method almost performs perfect motion estimation, while Bergen’s result is not as good. It is easy to interpret these trends in these results. Bergen’s method is based on the assumption of Brightness Constancy, thus it considers that the local/nonrigid motion in dynamic textures is also caused by camera motion. On the other hand, our approach is based on the more accurate sparse representation constancy assumption and seeks the optimal estimation in terms of all preceding image frames via $l_1$ minimization.

Table 6.4: FEF of horizontal cumulative motion

<table>
<thead>
<tr>
<th>Generated Sequence</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bergen’s [15]</td>
<td>22.55%</td>
<td>21.86%</td>
<td>23.25%</td>
</tr>
<tr>
<td>Proposed</td>
<td>1.39%</td>
<td>1.48%</td>
<td>1.36%</td>
</tr>
</tbody>
</table>

For quantitative evaluation, the false estimation fraction (FEF) is used to indicate the difference between the ground-truth $M_{True}$ and the estimated motion $M_{Est}$: $FEF = |M_{Est} - M_{True}| / |M_{True}|$. Table 6.4 records the FEF of registration results on the three generated image sequences by different implementations. Our algorithm gives very good motion estimation results.

The moving flower bed sequence shown in Figure 6.14 has been used as a registration example.

Figure 6.13: The cumulative motion estimation
The whole sequence includes 554 image frames. The camera motion in this sequence is a horizontal translation. The ground truth of the cumulative horizontal motion in the whole sequence is 110 pixels based on manual motion labelling of one red flower. While quantitative motion estimation results were not reported in [40], the FEF of the cumulative motion is 29.41% by Vidal’s approach on a subsequence with 250 frames [135]. The FEF of the cumulative motion on this sequence is reported as 1.7% in [106] and 4.98% in [58]. For quantitative comparison with these methods, we also tested the proposed algorithm on this sequence. The cumulative motion along the horizontal direction is estimated as 107.7 pixels by our approach, thus a 2.09% FEF of cumulative motion; this performance is close to the reported result by the extrapolation based registration approach [106]. Based on the efficient $l^1$ minimization, our algorithm takes less than 5 seconds to register each image frame on a 1.5GHz laptop PC in MATLAB environment, which is faster than extrapolation based registration using block-based video synthesis. It further demonstrates the effective performance of the proposed approach. As introduced above, the registration is based on the block-based video synthesis method, which is generally not fast. On the other hand, based on the efficient $l^1$ minimization, our algorithm takes less than 5 seconds for each image frame on a 1.5GHz laptop PC in MATLAB environment.

![Figure 6.14: A sequence of a moving flower bed [40, 106, 135, 58].](image)

6.3.5 Discussions

In this section, we extend the sparse representation to be invariant to a desired group of image-plane transformations of an ensemble of unaligned images. By coupling the recently emerged theories on compressive sensing and random projection manifold, the proposed approach can efficiently recover not only sparse representation of a target image but also the image plane transformation between the target and the model images. All of the above experimental results have validated the proposed transformation invariant sparse representation algorithm.

1. The sparse image representation, successfully extended to be invariant to a desired group
of image-plane transformations, is easily applied to image analysis related problems.

2. The sparse representation constancy assumption, proposed in place of brightness constancy assumption for motion estimation, has been validated and improves performance.

3. Compared to previous algorithms, the proposed algorithm can handle less constrained cases and promises better performance at the cost of more memory usage (3 and 7 times more for translation and affine transformations, respectively).

6.4 Summary

This chapter introduces how sparsity techniques could be used in the fields of medical image analysis and computer vision. We have proposed a sparsity driven method for tag separation in tMRIs, which can directly decomposed a tagged cardiac MRI to a tag-only image and another cardiac image without tags. The sparsity techniques are also used for cervigram image segmentation based on sparse coding and sparse recovery. We proposed classifiers based on reconstructive sparse representations to segment tissues in optical images of the uterine cervix. The method was compared with some typical learning methods and worked better in this challenging case. Finally, we also extend the sparse representation to be invariant to a desired group of image-plane transformations of an ensemble of unaligned images. The transformation-invariant sparse representation has been used for robust face recognition and dynamic scene registration and obtained far better performance.
In this thesis, we generalized the standard sparsity and group sparsity to structured sparsity and build a unified framework to incorporate the structure information into sparsity models. This new framework is general and flexible enough, which can be used in different fields, such as compressive sensing, sparse learning, computer vision and medical imaging. The main contributions of this thesis are summarized as follows:

• Group Lasso is a well-known algorithm for group sparsity in statistical learning. We have developed a theory for Group Lasso using a concept called strong group sparsity and theoretically prove that group Lasso is superior to standard Lasso for strongly group-sparse data. This provides a convincing theoretical justification for using group sparsity regularization when the underlying group structure is consistent with the data. Moreover, our theory can also help to predict some limitations of the group Lasso formulation.

• We have generalized the group sparsity idea to a new concept structured sparsity by allowing arbitrary structures on the feature set. A general theory Group-RIP is developed for learning with structured sparsity, based on the notion of coding complexity associated with the structure. It is shown that if the coding complexity of the target signal is small, then one can achieve improved performance by using coding complexity regularization methods, which generalizes the standard sparsity regularization.

• We have successfully applied the new framework of structured sparsity to practical applications on compressive sensing and computer vision. A new algorithm DGS is developed for compressive sensing. The proposed algorithm can reliably recover sparse data with clustering trends using far fewer measurements and computations than current state-of-the-art algorithms with provable guarantees. We also develop a new algorithm AdaDGS. It can adaptively learn the data structure and the sparsity number if they are not available in the practical applications. The proposed algorithm obtains superior performance over other state-of-the-art algorithms in background segmentation and visual tracking.
• We have proposed an efficient algorithm for the compressed MR image reconstruction. First, the proposed FCSA can efficiently solve a composite regularization problem including both TV term and $L_1$ norm term, which can be extended to other medical image applications. Second, the computational complexity of the proposed FCSA is only $O(p \log(p))$ in each iteration where $p$ is the pixel number of the reconstructed image. It also has strong convergence properties. These properties make the real compressed MR image reconstruction much more feasible than before. In numerous experiments, our method is shown to impressively outperform the classical methods and two of the fastest methods so far in terms of both accuracy and complexity.

• We have successfully applied the sparsity techniques to several practical applications in image processing and computer-aided diagnosis. First, we have proposed a sparsity driven method for tag separation in tMRIs, which can directly decompose a tagged cardiac MRI to a tag-only image and another cardiac image without tags. It obtained superior performance over all previous methods for this task. We also developed a new method based on sparsity techniques for segmenting tissues in optical images of the uterine cervix. The proposed method worked better in this challenging case compared with some typical methods in this application. Finally, we also developed a new algorithm for transformation-invariant sparse representation, which has been successfully used for robust face recognition and dynamic scene registration to obtain far better performance.

Our work also reveals the opportunity for a number of possible improvements and new research directions to further our understanding and applications of structured sparsity on machine learning, biomedical imaging, computer vision and compressive sensing. Some of the most interesting aspects are:

1) Accelerating MR Imaging Based on Structured Sparsity and Parallel Imaging

Compressive sensing can accelerate MRI with acceleration factor $3 \sim 5$. Parallel imaging can also make accelerations and its acceleration factor is limited by noise amplification. It is interesting to combine both of them to accelerate MRI under an unified framework. Applied to first-pass cardiac perfusion MRI, this combination will make it relatively insensitive to respiration motion artifacts. This can be a long-term project. I intend to continue this project and see if 10-fold acceleration can be achieved based on combination of distributed compressive sensing and parallel imaging.

2) Online learning with structured sparsity for huge scale data
Despite its success in many applications, current structured sparsity is optimized in batch-mode. In real-world application, the training data may appear sequentially or be huge scale. Online learning is a better choice to address these problems (Sequential data, poor scalability). On the other hand, surprisingly, the existing online learning literature overlooks the accelerated optimization on structured sparsity data. I am excited about developing theoretical bounds and algorithms for online learning with structured sparsity, which will open its application door on huge size data in practical applications.

3) Applications on Feature Selection

Standard sparsity model has been widely used for feature selection based on the assumption that useful features are few and independent. If graph structures exist in features, we can use our structured sparsity to take advantage of such structures to improve the standard feature selection algorithms and obtain better results. There are a few potential applications of structured sparsity models on feature selection:

- **Automatic Image Annotation**
  The extracted features from images can be represented by a graph with finite degrees. With these built graphs in features, we can use the structured sparsity models to select distinct features for image annotations with improved performances.

- **Medical Image Segmentation**
  The extracted features from patches in medical images can be represented by a graph with finite degrees. With these built graphs in features, we can use the structured sparsity models to select distinct features for image patch classifications with improved performances, such as tumor segmentation, cardiac segmentation and so on.

- **Robust Visual Tracking**
  In visual tracking, we generally scan a square windows in the current image frame to localize the interested objects. However, the interested objects are spatial sparse and the pixels of objects are also inclined to clustered. Therefore, the structured sparsity model can be well used to select the useful features (pixel) in the scanned window for matching in the tracking.

4) Applications on Anomaly Detection

Standard sparsity model can be used for anomaly detection based on the assumption that abnormality is intrinsically sparse. If graph structures exist in features, we can use our structured sparsity model to take advantage of such structures to improve abnormal detection and
obtain better results. There are a few potential applications of structured sparsity models on abnormal detection:

- **Robust Shape Alignment (Metrics/Occlusions)**
  Occlusions lead to difficulty for shape alignment. The structured sparsity can be used to well model the occlusions since the occlusions are not only sparse but also clustered. We can directly optimize the alignment parameters and occluded shapes to obtain robust alignment results.

- **Robust Image Registration (Metrics/Occlusions)**
  Occlusions lead to difficulty for image registration. The structured sparsity can be used to well model the occlusions since the occlusions are not only sparse but also clustered. We can directly optimize the registration parameters and occluded images to obtain robust registration results.

- **Video Decomposition**
  When the cameras are approximately fixed in video surveillance, the captured videos can be well decomposed into two videos, the foreground videos and the background videos. The background video can be well modeled by a low-rank matrix, while the foreground video can be well represented by the structured sparsity data. With these two models, we can directly decompose the video into two video by convex optimization.
References


